



UNIVERSITAT DE
BARCELONA

Treball final de grau

GRAU D'ENGINYERIA
INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

INTEGRACIÓ DE SERIOUS GAMES EN UN LMS

Autor: Raúl Vega Nielfa

Directora: Dra. Anna Puig

Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 27 de juny de 2019

Índex

1	Introducció	1
1.1	Àmbit del projecte	1
1.2	Motivació	1
1.3	Objectius generals	1
1.4	Objectius específics	2
1.4.1	Presa de requeriments	2
1.4.2	Anàlisi elecció LMS i protocol de comunicació	2
1.4.3	Anàlisi, disseny i implementació API comunicació en el LMS	2
1.4.4	Anàlisi, disseny i implementació API Moodle - Javascript . .	2
1.4.5	Anàlisi, disseny i implementació API Moodle - Unity	3
1.4.6	Implementació tres mini serious games en un LMS	3
1.5	Planificació temporal	3
1.6	Organització de la memòria	4
2	Anàlisi	5
2.1	Requeriments funcionals	5
2.2	Requeriments no funcionals	12
3	Disseny	15
3.1	Justificació de la tecnologia utilitzada	15
3.2	Arquitectura del sistema	16
3.2.1	MOODLE	17
3.2.2	Què és SCORM?	18
3.2.3	Runtime service	19
3.2.4	Comunicació	20
3.3	API Javascript	22
3.4	API Unity	23
4	Resultats i Simulacions	25
4.1	Math Quiz	25
4.1.1	Explicació joc	25
4.2	Math Quiz a Moodle	26
4.3	Breakout	27
4.3.1	Explicació joc	27

4.3.2	Serious Game Breakout	29
4.3.3	Serious Game Breakout amb Moodle	29
4.3.4	Explicació joc	30
4.3.5	Serious Game Space Shooter	32
4.3.6	Serious Game Spacer Shooter amb Moodle	35
5	Conclusions i feina futura	38
A	Manual tècnic	40
A.1	Instal·lació: Requeriments mínims i passos a seguir	40
A.2	Manual del desenvolupador: afegir un nou joc	41
A.2.1	Moodle	41
A.2.2	Javascript	45
A.2.3	Unity	45
B	Manual d'usuari	46

Abstract

Recently, usage of management tools and online communication in the world of learning has become a new breakthrough for teachers and students while doing a simple job tracking. The monitoring of some of these tools has had a great impact on society, such was the case that many institutes and universities use it. Currently, the pedagogical focus on the use of new methodologies such as the serious games has described improvements in students who have opted for more than one apprenticeship with one of these tools, which a learning more conventional following a book and problems.

For these reasons, this project seeks to verify the integrated use of these two tools. On one hand, the use of the Learning Management System, or LMS, keeping in mind that the roles of users, teachers and students, and the management of courses and activities. On the other hand, the integration of one or more serious games in a Learning Management System website platform . At this point, keep in mind the correct choice of a simple communication protocol between LMS and serious games.

Resum

Des de fa ja un temps, la utilització d'eines de gestió i comunicació online en el món de l'aprenentatge ha esdevingut un nou avanç per professors i alumnes alhora de realitzar un seguiment senzill de la feina. El seguiment d'algunes d'aquestes eines ha tingut un gran impacte en la societat, tant és així que molts instituts i universitats l'utilitzen. En l'actualitat, l'enfoc pedagògic en l'ús de noves metodologies com els serious games ha denotat millores en estudiants que han optat més per un aprenentatge amb una d'aquestes eines que un aprenentatge més cnvencional seguint un llibre i problemes.

Per aquestes raons, en aquest projecte es busca comprobar l'ús integrat d'aquestes dues eines. Per una banda, l'utilització del Learning Management System, o LMS, tenint en compte els rols d'usuaris, professor i estudiant, i la gestió de cursos i activitats. I per una altra, la integració d'un o varis serious games en una plataforma de Learning Managment System. En aquest punt, s'ha de tenir en compte l'ús d'escollir un bon protocol de comunicació entre el LMS i els serious games.

Índex

1 Introducció

1.1 Àmbit del projecte

Aquest projecte s'emmarca dins de l'àmbit de l'educació, on professors i alumnes utilitzaran una eina bastant coneguda aquests últims anys com és un LMS (Learning Management System). Aquesta eina és una plataforma web que té funcionalitats per a professors com la de pujar contingut d'estudi, pujar activitats que realitzaran els alumnes, petits tests o fòrums per discutir temes, entre d'altres. Pels alumnes, el LMS facilita els processos d'aprenentatge d'una manera més dinàmica i àgil, podent tenir apunts d'una manera directa des de casa, agilitzant les proves telemàtiques, afegint la possibilitat de fer xats amb altres companys per resoldre dubtes, pujar exercicis, feina o pràctiques que el professor demana, etc.

Alhora de profunditzar en metodologies pedagògiques, els *serious games* són eines que permeten als alumnes aprendre d'una manera divertida, i per tant més relaxada sense tenir la pressió de pensar que has de llegir-te un text o has d'anar a una classe on has d'estar centrat. Un serious game s'entén com un joc on la seva finalitat principal és l'aprenentatge mitjançant el joc. D'aquesta manera, aconseguir integrar un serious game en una eina e-learning com és un LMS, donaria un aire fresc i/o una nova visió en l'àmbit educacional.

1.2 Motivació

L'impacte que pot tenir un serious game en un entorn com és el dels LMS pot ser interessant i potent alhora de gestionar les habilitats i l'aprenentatge sobretot de alumnes en edat escolar, ja que per ells jugar és una manera més divertida i senzilla d'aprendre. També podria ser interessant per persones majors d'edat que estan treballant en una empresa que els faci reciclar fent cursos, on serious games de tipus quiz poden ser efectius enfocar-los perquè siguin un repte de superació.

Per aquesta raons, es veu la necessitat de gestionar un LMS en el qual s'integraran un parell de serious games, a mode de prova. El projecte pot ser un repte ja que a més a més d'aconseguir fer que funcioni un joc en un entorn com el de un LMS, cal aconseguir que es comuniqui amb aquest. La comunicació ha de ser viable i s'ha de poder enviar dades des del LMS tals com paràmetres per configurar el serious game, i també s'ha de poder rebre dades des del serious games, tal com pot ser la puntuació del joc i rebre-la en el LMS en forma de taula de resultats.

1.3 Objectius generals

L'objectiu general de projecte és definir el procés d'incloure serious games en un LMS. Concretament, cal definir una API que permeti integrar en el LMS jocs desen-

volupats en diferents tecnologies com són JavaScript o Unity 3D. Així cal poder incloure en un LMS un serious game desenvolupat en Javascript sense GUI (Graphic User Interface), un altre també amb Javascript i amb GUI, i un tercer en Unity amb GUI. El LMS ha de tenir en compte el rol de professor que podrà configurar uns paràmetres bàsics que es guardaran en una taula que es crearà en la base de dades del LMS, després haurà de visualitzar la puntuació obtinguda de l'alumne en una taula. També s'haurà de tenir en compte el rol de l'alumne que jugarà al serious game i tindrà el repte de, a la vegada que va sortejant els diferents obstacles d'un joc, anar practicant càlculs matemàtics, i on el càlcul final de la seva puntuació tindrà més que veure amb els càlculs ben fets, que amb la seva destressa com a jugador.

1.4 Objectius específics

1.4.1 Presa de requeriments

Primerament, s'es definiran els requeriments que necessita el projecte en relació amb el LMS. En quant als requeriments funcionals es definirà què necessita cada usuari, que dependrà del rol que jugarà, i quines opcions disposarà perquè sigui possible la comunicació entre el LMS i el serious game. Pels requeriments no funcionals s'hauran d'identificar quin tipus d'infraestructura és adient per aquest tipus de LMS i quin tipus de software o hardware mínim cal tenir instal·lat.

1.4.2 Anàlisi elecció LMS i protocol de comunicació

En aquest objectiu cal definir els requeriments a complir pel LMS, analitzar els que estan disponibles i són més utilitzats, i decidir quin servirà de base per aquest projecte. També cal estudiar el mateix amb els protocols o estàndards de comunicació i definir el protocol a utilitzar en el projecte.

1.4.3 Anàlisi, disseny i implementació API comunicació en el LMS

En aquesta part s'ha de investigar com funciona l'arquitectura de comunicació entre el LMS i el serious game, ja que és el punt crític del que es vol fer en aquest projecte. S'hauria de saber les parts que conforma aquest estàndard i explicar-les, a més a més de explicar com funcionen els mètodes principals. Explicar com inicia la comunicació, com envien i rep dades el serious game i com finalitza la comunicació.

1.4.4 Anàlisi, disseny i implementació API Moodle - Javascript

En aquest objectiu cal analitzar, dissenyar i implementar l'API de comunicació en un joc on el llenguatge principal és el Javascript. Caldrà definir quines funcions cridarà el joc per comunicar-se o en quin moment es fan les crides.

1.4.5 Anàlisi, disseny i implementació API Moodle - Unity

Per aquest objectiu cal analitzar, dissenyar i implementar l'API de comunicació en un joc on l'interfaç es construirà en Unity i el llenguatge és el C sharp. Caldrà definir en quines parts del llenguatge de programació que s'usa a Unity s'implementaran les funcions de comunicació per enviar la informació al LMS.

1.4.6 Implementació tres mini serious games en un LMS

Finalment, s'implementaran tres serious games en el LMS. Aquests serious games seran un joc en Javascript sense GUI, un altre joc en Javascript amb GUI, i un tercer joc en Unity amb GUI. Quin control té un professor per donar unes opcions a la configuració del serious game o com l'alumne interactuarà amb el serious games són temes que s'explicaran en aquest punt.

1.5 Planificació temporal

Tasques principals del projecte

Presa de requeriments Es mencionaran els requeriments necessaris per assolir els nostres objectius. Seran necessaris uns requeriments funcionals on es mencionarà tot el que necessita el nostre projecte, a més d'uns no funcionals on s'esmentaran tota classe de software, hardware i requeriments humans.

Anàlisi i elecció LMS i protocol de comunicació S'analitzaran els LMS que existeixen i es durà a terme un estudi exposant els avantatges i inconvenients generals dels més coneguts i amb més presència a la xarxa per finalment escollir un. El mateix amb el protocol de comunicació, tenint en compte l'elecció del LMS.

Anàlisi, disseny i implementació API Moodle - Javascript Es durà a terme una anàlisi i estudi de l'API de comunicació entre el LMS i la connexió amb Javascript per conèixer quina és la millor manera d'implementar-la.

Anàlisi, disseny i implementació API Moodle - Unity Es durà a terme un anàlisi i estudi de l'API de comunicació entre el LMS i la connexió amb Unity, caldrà entendre quina alternativa a Javascript caldrà seguir per connectar aquest API amb LMS i trobar la millor implementació.

Implementació tres mini serious games en un LMS S'integraran tres serious games en el LMS escollit. Caldrà saber quina manera té el servidor de gestionar la persistència de dades per després enviar-la als serious games.

Ja mencionats les nostres tasques principals, caldrà veure com planificar el projecte. La planificació de la figura 1 que s'ha estimat, es durà a terme en aquest

projecte és una metodologia àgil, com es la de SCRUM, a més a més es tornaran a revisar els requeriments per veure si es pot enfocar d'un altre forma o millorar-los, per tant el nostre SCRUM serà cíclic i iteratiu.

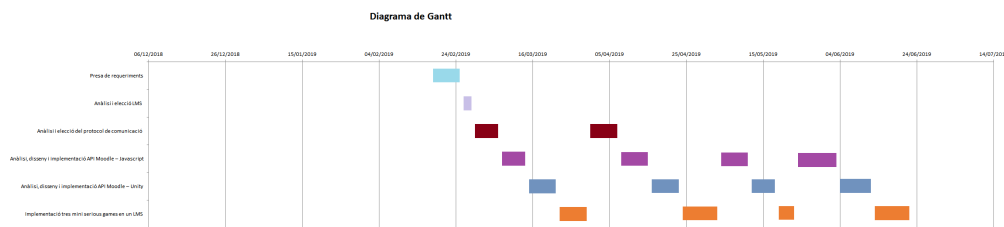


Figura 1: Diagrama de Gantt

1.6 Organització de la memòria

La memòria d'aquest projecte es dividirà en quatre apartats a més a més d'aquesta introducció.

Primerament, començarem amb un anàlisi de l'aplicació on es detallaran els requeriments funcionals i els no funcionals. S'utilitzaran imatges per detallar les pantalles o fases que els usuaris necessitaran al llarg del seu recorregut pel LMS.

Una vegada feta l'anàlisi, s'enfocarà a detallar el disseny de l'aplicació. En aquest capítol, es presenta la justificació de la tecnologia utilitzada en l'aplicació, esmentant pros i contras per a acabar donat la solució escollida. Després s'explica l'arquitectura de les tecnologies elegides, com funcionen internament. Per acabar, el capítol parla de com aplicar aquestes tecnologies en cadascuna dels serious games que es volen integrar.

El tercer capítol presenta els resultats obtinguts i les simulacions dels serious games en l'entorn LMS.

Per acabar, en l'últim apartat es presenten les conclusions i feina futura.

Es presenta també l'annex del manual del desenvolupador, així com un annex amb el manual d'usuari

2 Anàlisi

Per saber el que es vol oferir amb aquest projecte, cal detallar l'abast d'un LMS. LMS [3] és l'acrònim anglès de Sistema Gestor d'Aprenentatge. Un LMS proporciona cursos i material d'aprenentatge online gestionats entre els participants que no són més que els usuaris amb diferenciats pels seus rols, professor - alumne. Aquest tipus de plataforma està enfocada a escoles, instituts o empreses. És una manera fàcil de no tenir que carregar amb llibres, sinó que ja es disposa de tot en la plataforma.

En aquest capítol s'expliquen els requeriments del problema, funcionals i tecnològics. En relació a l'extensió de les activitats del LMS (*Learning Management System*), cal poder:

- configurar diferents usuaris.
- configurar el joc des del LMS
- poder executar el joc des del LMS
- poder recollir els resultats des del LMS per integrar-los a les dades de l'estudiant.

Un requeriment bàsic del LMS és organitzar la informació de manera que dependent del rol dels usuaris, puguin accedir, consultar i modificar dades.

2.1 Requeriments funcionals

Per explicar el procés que utilitzarà cada usuari i les funcions que farà, es dissenyarà un prototipus base. A continuació s'enumeren les diferents pantalles del *story board* que definiran les interaccions amb els diferents perfils d'usuari.

En primer lloc, tot usuari s'haurà d'identificar en el LMS i així podrà tenir els privilegis o les possibilitats de fer operacions en el LMS. Normalment, els rols en un LMS són el rol de professor i el rol d'estudiant, a més del rol d'administrador, propi del gestor del sistema. Per aquesta raó, el primer que es visualitzarà en l'entorn del LMS és una finestra de Login (veure Figura 2).

Una vegada s'hi iniciï sessió, l'usuari accedirà a la pantalla principal del seu compte (Figura 3). Normalment, en aquesta pantalla es trobaran les diverses opcions típiques del LMS, cursos, calendari, blocs comuns de totes les pàgines d'inici, etc.. Aquest projecte es focalitzarà en cursos, on els alumnes i professors podran interactuar amb les diverses opcions d'aquests.

Tant el login com la pantalla principal seran pantalles comuns a tots els tipus d'usuaris del LMS. Serà a partir d'aquest punt on difereix les següents pantalles, ja que uns usuaris podran accedir a un lloc on l'altre no.

Administrador/Professor

LMS

Learning Managment System

Ja tens usuari registrat al LMS ?

Login

Username

Password

Loguear

No tens usuari registrat? [Demana accés aquí](#)

Figura 2: Pantalla inicial de login al LMS.

LMS

Learning Managment System

Nom d'usuari

Benvingut al teu espai del Learning Managment System

Página de inici

Calendari

Els meus cursos

Curs 1

Curs 2

Curs 3

Curs 1

Curs en progrés

25% començat

25% no començat

50% acabat

Bloc 1

Bloc 2

Bloc 3

Figura 3: Menu principal del LMS.

La primera acció que farà un professor serà donar d'alta una activitat en el seu curs i aquesta activitat serà de tipus serious game. Per això, primer caldrà habilitar l'edició del curs per poder afegir una activitat on s'ubicarà el joc. El requeriment principal d'aquest tipus d'activitat haurà d'utilitzar un estàndard o protocol general que pugui habilitar la comunicació entre el serious game i el LMS, per això s'elegirà un tipus d'activitat SCORM [2] o TinCan [5], ambdues d'ora en endavant eines estàndard o mecanismes que podran donar base en la comunicació del LMS.

SCORM és un estàndard que ja porta temps en el mercat, aproximadament un 25-30 anys, aquest protocol donarà una funcionalitat senzilla, amb no massa paràmetres i funcionarà bastant bé en LMS coneguts. La API de TinCan és un estàndard que no porta molt de temps en la xarxa, ha arribat per revolucionar l'experiència de l'aprenentatge i pot capturar events tant online com offline en eines d'aprenentatge.

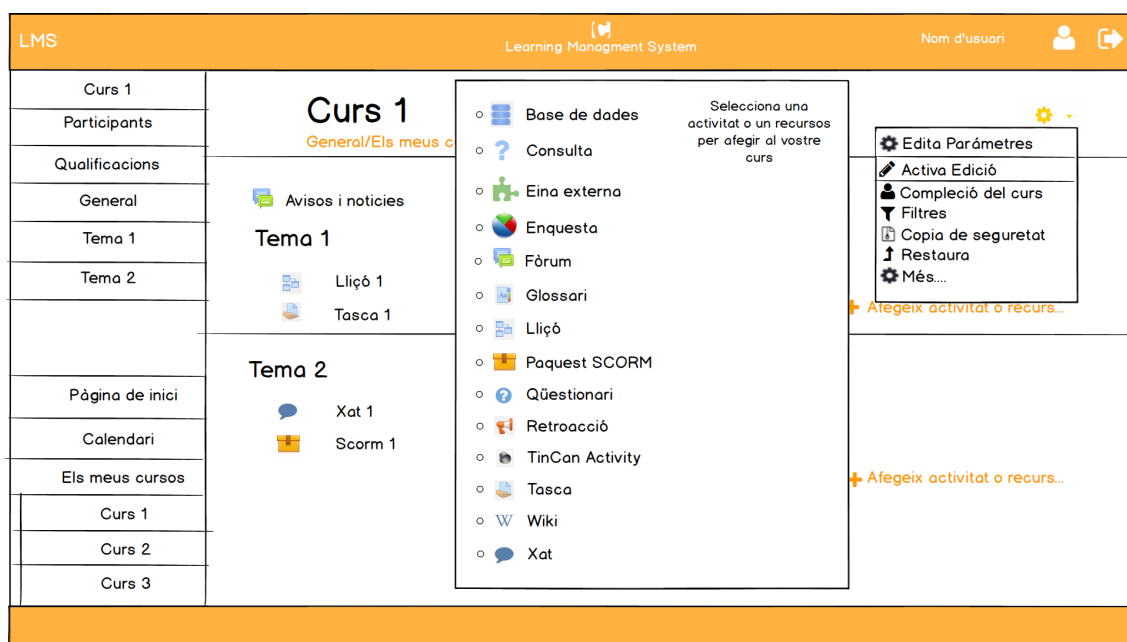


Figura 4: Habilitació per crear activitat.

Una vegada s'obri la pantalla de configuració de l'activitat (veure Figura 4), es configuraran tres paràmetres bàsics tals com el nom, una descripció i un fitxer que servirà per pujar el joc, per defecte en format .zip o.xml. Tots tres paràmetres seran obligatoris ja que el LMS no deixarà salvar l'activitat sinó seran emplenades. El joc es podrà pujar utilitzant el botó de pujada com es pot veure a la Figura 5), o arrossegant-lo i deixant-lo anar dins del espai on es pot veure la fletxa. L'enllaç de la configuració es farà mitjançant el fitxer .xml o dins del .zip ha de existir aquest fitxer en primera plana. A partir d'aquí, es faran les crides pertinents a fitxers que executaran el flux del joc.

Una vegada s'ha acabat de configurar l'activitat del joc, existiran tres botons. El botó **Cancel·lar** servirà per a sortir de l'activitat sense guardar els canvis. El segon botó serà **Guardar i tornar al curs, sense provar el joc**, i l'últim botó de guardar i visualitzar, aquí si que es podrà jugar i testear l'activitat directament.

LMS		Learning Management System		Nom d'usuari	
Curs 1	Curs 1				
Participants	General/Els meus cursos/Curs 1/Afegir activitat				
Qualificacions	Afegint nova activitat				
General	<div> <div>Opcions generals</div> <div> <div>Nom</div> <input type="text"/> </div> <div> <div>Descripció</div> <div></div> </div> </div> <div> <div>Fitxer</div> <div>Fitxer paquet</div> <div> <input type="text"/> <div>Arrossega l'arxiu i deixa'l anar per afegir-lo</div> </div> </div>				
Tema 1					
Tema 2					
Pàgina de inici					
Calendari					
Els meus cursos					
Curs 1	<div> <div>Guardar i tornar al curs</div> <div>Guardar i visualitzar</div> <div>Cancelar</div> </div>				
Curs 2					
Curs 3					

Figura 5: configuració activitat per pujar joc

El rol d'administrador ja no tindrà res més a fer, la configuració de l'activitat, com he esmentat abans, es feina d'un administrador o un professor-administrador. A partir d'aquí el flux es dividirà pels dos rols (Professor i estudiant). Cadascú, farà un ús diferent de l'activitat.

Estudiant

Una de les opcions que es tindrà des de la vista alumne serà accedir a l'activitat del joc (veure Figura 6) . L'usuari jugarà a un serious game que s'haurà de comunicar mitjançant el arxiu .xml on s'anirà a configurar, aquest fitxer farà la crida principal a un fitxer .html, i aquest en farà una de nova al Javascript per iniciar el joc. El primer que es voldrà aconseguir es que funcioni la comunicació entre el LMS i el joc, ja que es necessitarà que es facin els **sets** i **gets** necessaris perquè funcioni el joc. Després es podrà jugar al joc. El serious game combinarà habilitat a la vegada que l'alumne haurà de fer servir la lògica per realitzar càlculs (sumes, restes, multiplicacions i divisions).

Una vegada s'acabi el joc, o es surt, aquest guardarà automàticament resultats de **score**, i comprovarà si s'ha fet màxim o mínim record. Per acabar, el resultats es visualitzaran en una pantalla al sortir de l'activitat del joc (veure Figura 7), aquesta pantalla serà la de informes dels resutats del joc i mostrarà paràmetres tal com del score, score màximo o mínim per exemple.

Evidentment, l'alumne podrà jugar tantes vegades com vulgui al joc. Fins que l'activitat s'oculti per un administrador o professor o hagi vençut.

Professor

Les vistes que tindrà el professor seran, per una banda la de configurar els paràmetres del joc, i per l'altre veurà els Informes de tots els alumnes que han

tingunt contacte amb el joc.

La vista de configuració de les opcions del joc (veure Figura 8) es basarà en tres paràmetres. El primer paràmetre serà un número, el qual en el joc servirà per fer els càlculs pertinents en aquest. Evidentment, com més gran sigui el número més dificultat tindrà per l'alumne realitzar els càlculs, per aquesta raó s'ha configurat perquè aquest número sigui entre 1 i 10.

Figura 8: Opcions des de la vista del professor

El segon paràmetre a configurar serà el signe (suma, resta, multiplicació o divisió). La elecció només depndrà d'un radio-button, aquest paràmetre servirà en el joc per elegir quin tipus de càlcul es realitzarà en el moment en que l'alumne estigui jugant.

El tercer paràmetre que es configurarà serà la dificultat. Per aquest paràmetre hi hauran tres eleccions possibles (fàcil, moderat, difícil). En el joc es podrà visualitzar aquesta elecció amb els números que ens trobarem, com més difícil siguin els números, més gran sortiran al joc per calcular.

La vista dels informes dels professors estarà basada en una taula on seran recollits tots els informes dels alumnes (veure Figura 9). En aquesta taula es visualitzarà per cada fila un alumne i on cada columna serà una dada recol·lectada en el joc, tals com el score, score màxim o mínim.

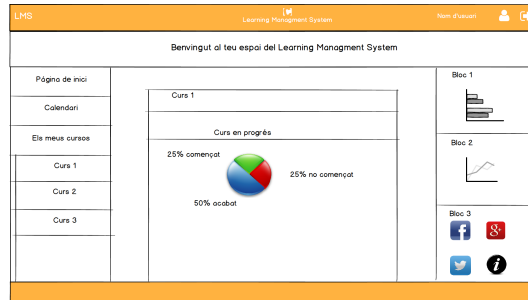


Figura 11: menu principal LMS

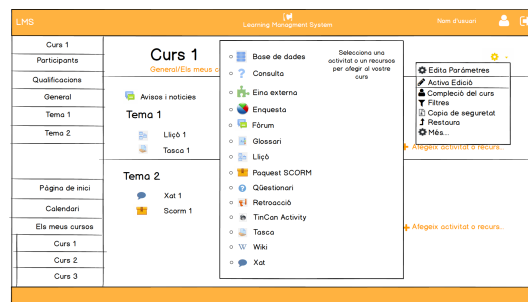


Figura 12: Curs seleccionat

El professor configurarà l'activitat i afegirà l'arxiu amb el joc. (veure Figura 13)

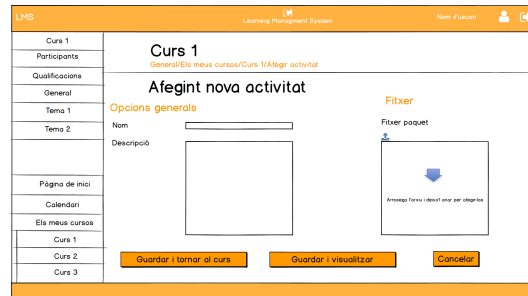


Figura 13: nova activitat pel joc

El professor configurarà els paràmetres del joc dins de l'activitat que s'ha creat. (veure Figura 14)

L'alumne jugarà dins de l'activitat del joc. (veure Figura 15)

L'alumne veurà l'informe de la seva partida. (veure Figura 16)

El professor podrà veure informes dels alumnes que han jugat. (veure Figura 17)

2.2 Requeriments no funcionals

Els requeriments hardware i software que es necessitaran per l'aplicació seran un ordinador convencional pels usuaris, i un servidor per allotjar el LMS. No caldrà

LMS Learning Management System Nom d'usuari

Curs 1

Participants

Qualificacions

General

Tema 1

Tema 2

Pàgina de inici

Calendari

Els meus cursos

Curs 1

Curs 2

Curs 3

Curs 1

General/Els meus cursos/Curs 1/activity resource

Propietats del joc

Número:

Signe: ☐ Suma ☐ Resta ☐ Multiplicació ☐ Divisió

Dificultat: ☐ Fàcil ☐ Moderat ☐ Difícil

Guardar **Cancel·lar**

Figura 14: Configuració joc

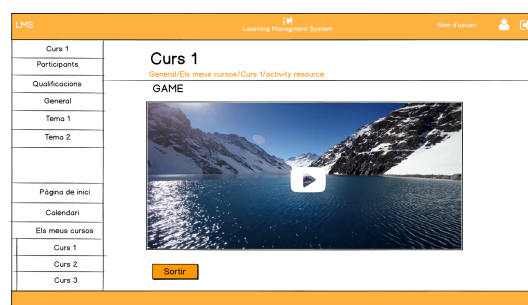


Figura 15: joc

LMS Learning Management System Nom d'usuari

Curs 1

Participants

Qualificacions

General

Tema 1

Tema 2

Pàgina de inici

Calendari

Els meus cursos

Curs 1

Curs 2

Curs 3

Curs 1

General/Els meus cursos/Curs 1/activity resource

Informes

Nom i Cognom	Puntatge	Percentatge
Jordi	10	100%
Marta	10	100%
Pau	10	100%

Entrene

Figura 16: Informe estudiant

LMS Learning Management System Nom d'usuari

Curs 1

Participants

Qualificacions

General

Tema 1

Tema 2

Pàgina de inici

Calendari

Els meus cursos

Curs 1

Curs 2

Curs 3

Curs 1

General/Els meus cursos/Curs 1/activity resource

Informe Resultats Alumnes

Nom i Cognom	Puntatge	Percentatge
Jordi	10	100%
Marta	10	100%
Pau	10	100%

Entrene

Figura 17: Informes dels estudiants

tenir cap programa addicional instal·lat, amb winzip (o programa semblants) i un navegador que pugui fer servir el Javascript.

Fora de l'aplicació, es necessitarà d'una infraestructura que suporti diferents tipus d'usuaris, ja que l'aplicació necessitarà almenys tres usuaris amb rols diferents perquè funcioni, tal i com he esmentat abans serien els següents: administrador, professor i alumne.

Es necessitarà un sistema actiu les 24 hores del dia, durant tot l'any lectiu del curs que suporti l'infraestructura, i un servidor que garanteixi aquest funcionament.

El problema que s'exposa serà un sistema multiusuari, on diferents usuaris alhora poden entrar en la nostra aplicació, per tant es necessitarà una infraestructura que suporti aquesta característica.

Per arribar a aconseguir els objectius esmentats, es necessitarà un LMS que tingui una comunitat activa per poder preguntar dubtes que puguin aparèixer i/o poder solucionar-los.

3 Disseny

3.1 Justificació de la tecnologia utilitzada

Analitzar-lo en relació a altres possibles solucions.

La necessitat d'escollir un protocol o estàndard que comuniqui el nostre serious game amb el LMS ha fet que s'investiguin quines eines s'utilitzen en aquests àmbits. El primer que es focalitzarà és en escollir un LMS que s'adapti a les necessitats de l'aplicació, per després escollir el protocol de comunicació que es farà servir amb el serious game.

Hi ha uns quants LMS interessants en la xarxa, a continuació esmentarem uns dels més importants i/o interessants. Wordpress, BlackBoard, Chamilo LMS o Moodle. D'aquests es van estudiar quins avantatges i desavantatges tindrien pel nostre projecte. Wordpress pot donar solucions de LMS de bona qualitat i podria ser una bona opció, però té una limitació i és que no té l'abast d'activitats que tenen altres LMS tals com tallers d'evaluació. BlackBoard és un altre LMS que podria ser una bona opció a més a més crida l'atenció que tingui eines focalitzades per persones amb problemes de visió, tot i així el problema d'aquest LMS recau en la seva velocitat, amb caigudes del servidor, una interfície d'usuari difícil de seguir o que té problemes de seguretat. Chamilo LMS i Moodle serien bones opcions. Tot i això Chamilo LMS té un petit problema i és que a vegades no es tan personalitzable i sobretot amb alguns formats estàndards.

Per aquestes raons i els següents motius es va decantar la balança per **Moodle** ja que és el que dona millors prestacions.

- La infraestructura de Moodle és un sistema que soporta activitat durant tot el dia i té capacitat de soportar molts usuaris connectats alhora, a més més de connexió multiusuari alhora.
- Soporta molts tipus d'activitats (forums, chats, tasques, qüestionaris, enquestes, i també té mòduls de comunicació).
- Moodle compta amb una gran comunitat on pots consultar dubtes en els seus forums, i a més a més té una wiki on pots conèixer com funciona cadascun dels mòduls i blocs. Això fa que sigui el LMS molt extens i amb molts usuaris desenvolupadors.
- És el LMS més utilitzat per entitats públiques com instituts, universitats, que utilitzen tant professors com alumnes els recursos que dona aquesta eina, i també utilitzat a empreses com a recurs per reciclar els seus treballadors.

Una vegada escollit el LMS, cal investigar un protocol de comunicació que es pot utilitzar en aquest. Els dos protocols més utilitzats en l'actualitat són SCORM i TinCan (o xAPI), tots dos protocols per monitoritzar activitats d'aprenentatge. Ambdós proveeixen d'un framework i estructura a través del qual envia el contingut de l'aprenentatge, però l'extensió i flexibilitat difereixen significativament.

SCORM

SCORM [1] és l'acrònim en anglès de Sharable Content Object Reference Model és un conjunt d'especificacions o estàndards creant objectes enfocats a l'aprenentatge.

- És un estàndard de monitorització per LMS, llenguatge que ja porta temps en el món de la tecnologia.
- És l'estàndard més ampliament utilitzat en el món del elearning.
- No té manera d'abastar tota la informació que succeeix tant online com offline i en diferents àmbits de l'aprenentatge .
- A Moodle es pot crear una activitat SCORM fàcilment, ja que existeix un mòdul integrat.

TinCan

- Permet monitoritzar en més context, no tan sols en LMS.
- Permet gravar, monitoritzar, personalitzar, millorar de les activitats d'aprenentatge, tant online com offline.
- Més accessible a desenvolupadors i més robust.
- Per incloure una activitat TinCan cal instal·lar un mòdul que no és fàcilment instal·lable i de configurar.

Justificar l'elecció de SCORM davant de TinCan, etc.

Potser, TinCan hagués estat una tecnologia més completa perquè té més avantatges que SCORM com bé s'ha exposat anteriorment. Tot i així, SCORM té l'avantatge que funciona bé i que un LMS com Moodle, adapta el mòdul SCORM directament. A més a més és el model de referència més útil i és fàcil de generalitzar. Potser TinCan ofereix més elements que podrien enriquir la infraestructura, però s'ha d'instal·lar a Moodle, i no sempre funciona bé o com tindria que funcionar, mentre que SCORM és un mòdul ja en l'estàndard de Moodle, i després de ser testejat de diverses maneres és senzill d'utilitzar i el seu abast de proves és suficient en el marc d'aquest projecte.

3.2 Arquitectura del sistema

En aquesta secció es detallen els principals mòduls de l'arquitectura del sistema que s'ha dissenyat. En primer lloc es descriu l'entorn Moodle i les bases de dades que fa servir, les seves funcionalitats bàsiques. Després es detalla l'estàndard SCORM per finalment presentar l'arquitectura de comunicació que s'ha dissenyat per a comunicar via SCORM, els jocs desenvolupats.

3.2.1 MOODLE

Moodle ¹ (veure Figura 18) és un sistema gestor d'aprenentatge (Learning Management System) o un entorn virtual d'aprenentatge. L'objectiu de Moodle és donar eines a professors i alumnes per assolir les necessitats d'ensenyar i aprendre. Moodle va arribar com una eina social pedagògica, tot i que pot soportar qualsevol estil d'ensenyança o aprenentatge. Moodle intenta no reinventar altres softwares o aplicacions sobre sistemes que ja existeixen en la xarxa, sinó conviure amb altres sistemes però sent el millor LMS. El llenguatge utilitzat en Moodle és Php i és open-source.

La instal·lació de Moodle compren varies parts, tals com un codi de llenguatge Php pel servidor, una base de dades que pot ser dirigida per alguns dels més famosos gestors de bases de dades tals com (MySQL, Postgress, Oracle, MariaDB o SQLServer), i un arxiu necessari pel emmagatzema la metadata. Totes tres parts poden funcionar en un servidor o per separat. Moodle està especialment dissenyat per a que sigui fàcil d'instal·lar.

El servidor sobre el que s'ha instal·lat Moodle és un servidor XAMPP ². Aquest servidor inclou en la seva arquitectura les següents aplicacions: Apache Http Server, base de dades MariaDB i intèrprets escrits en php i perl.

La infraestructura bàsica de Moodle conté tots els programes addicionals necessaris per a donar les prestacions elementals d'un LMS. Això vol dir que inclou:

- **Cursos, activitats i recursos:** un curs de Moodle és un grup d'activitats agrupades en seccions. Les activitats i recursos són els components més bàsics i les principals eines utilitzades per ensenyar i aprendre.
- **Usuaris:** per participar en els cursos de Moodle els usuaris es divideixen en dos rols principals, professors o estudiants. Els usuaris s'han d'inscriure en els cursos amb rol de professor o alumne. També, existeix la figura d'administració, que té el control del sistema.
- **Rols i funcionalitats:** un rol pot donar la possibilitat a un usuari de participar en un curs. També, l'accés a certes funcionalitats de Moodle pot permetre a un rol a accedir a certs espais. Tot això depèn dels permisos que es donen a cada rol.

Aquestes són les funcionalitats més importats que disposa el nucli de Moodle, però, a part, cal esmentar d'altres més com: accés a perfils d'usuari i edició, creació de grups de companys, compleció de cursos i activitats quan existeix certa condició, configuració de blocs que donen accés a certs llocs del LMS, ús de la llibreria Javascript, creació de backups i restauració d'aquests, logs i estadístiques i actualització a noves versions. També s'ha de tenir en compte que es poden instal·lar nous mòduls i nous blocs ampliant així les prestacions del LMS.

¹https://docs.moodle.org/dev/Moodle_architecture

²<https://en.wikipedia.org/wiki/XAMPP>

Base de dades: La base de dades de Moodle té més de 250 taules ³. Els programes o els plugins agregats acostumen a afegir taules al conjunt bàsic del nucli de Moodle. El model que utilitza Moodle de base de dades és un model relacional basat en els conceptes entitat-relació. La instal·lació de la base de dades és automàtica en la pròpia instal·lació de Moodle. La configuració es fa des de un fitxer .xml, normalment install.xml. En el cas de voler canviar una taula o taules en concret de la base de dades o afegir algun camp més en alguna taula, caldria desinstal·lar els mòduls concrets que tenen aquesta taula, canviar els paràmetres que es volen modificar en el .xml per després tornar-la a instal·lar.

Moodle té una part de la base de dades on guarda informació sobre SCORM. Aquestes taules són modificables si es volgués afegir-ne més. Això donaria més opcions alhora de monitoritzar noves dades i donar-li's persistència al LMS.

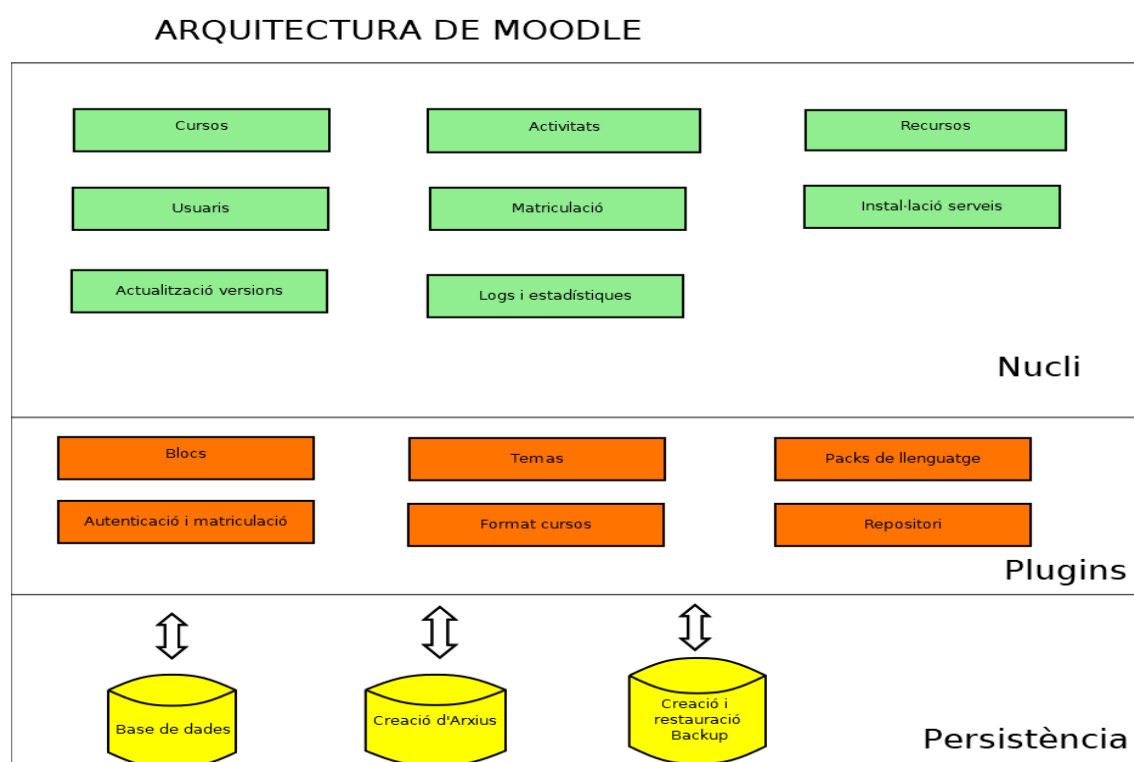


Figura 18: Arquitectura Moodle

3.2.2 Què és SCORM?

El significat de les sigles SCORM és **Sharable Content Object Reference Model**. És a dir, SCORM és un conjunt d'estàndards i especificacions amb finalitat pedagògica i àmbits educatius. La potència d'aquest estàndard esdevé en que es pot reutilitzar i compartir amb facilitat mitjançant gestors de contingut LMS. Cal tenir en compte que aquest estàndard només es pot comunicar a través del mateix

³https://docs.moodle.org/dev/images_dev/5/5a/Moodle2erd.png

navegador, en el costat del client i no seria possible usar-lo per a comunicar dos clients entre ells.

És a dir, aquest estàndard és el protocol per a comunicar un client i un servidor. Per a que la comunicació sigui possible entre client i servidor s'ha de cridar a l'API del SCORM en el costat del client, el runtime per aquesta raó el servei de execució (*runtime service*) ha d'instanciar l'SCORM API.

3.2.3 Runtime service

El runtime service [10] [11] pot ser implementat tant en la part servidor com en la client. Tot i que el treball està focalitzat en la part de client per que l'API de SCORM cal que s'engegui en el client i és on s'executaran tots els elements que necessitem. Per tant, en la part client del runtime service cal esmentar:

- Aquest servei està proporcionat pel LMS com una pàgina web o un marc principal en una finestra d'exploració.
- Atorga a l'usuari una interfície de components d'usuari (taules de contingut, botons de navegació).
- Introdueix objectes d'aprenentatge en una "stage window" que forma part del frameset o l'entorn integrat a la interfície gràfica de Moodle. També permet obrir finestres separades.
- Inclou un objecte instanciat de la API, que pot ser trobat fent la crida amb Javascript.

Per fer funcionar l'entorn d'execució en temps real hi han tres aspectes que s'han de tenir en compte i que es detallen en (veure Figura 19)

Launch/Execució :

L'execució del SCO, Sharable Content Object, és responsabilitat del LMS i per tant, té que tenir els mecanismes per poder entregar els recursos d'aprenentatge en forma de SCO i assets. El LMS pot implementar aquesta execució de diferents formes. També ha de complir les especificacions del protocol HTTP en l'enviament dels recursos. Normalment, el recurs d'aprenentatge està localitzat en el contingut del paquet, i es llençat i entregat al explorador client.

API :

És un mecanisme de comunicació que informa al LMS de l'estat dels recursos d'aprenentatge. Els diferents estats són d'aquests recursos serien execució, administració d'errors i transferència de dades. Aquesta envia i demana informació al LMS. El LMS proveeix de una API Adapter que implementa la funcionalitat de la API i mostra la seva interfície al client SCO.

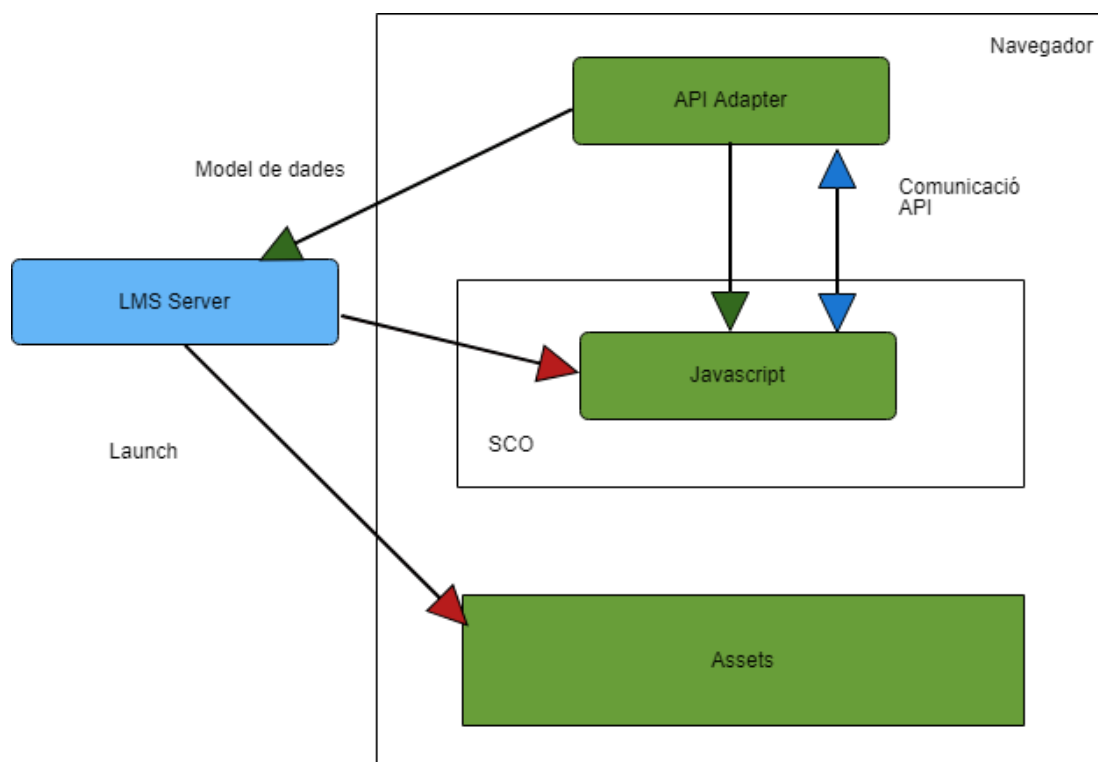


Figura 19: Entorn run-time proposat basat en SCORM

L'API Adapter és una instanciació que proporciona la API de Moodle mitjançant el run-time service al navegador client. El LMS ha d'executar el SCO en un navegador com a finestra fill del LMS que conté el API Adapter, aquest adapter ha de ser accessible via DOM. La interacció entre la API Adapter i el SCO ha de ser a partir de crides des de Javascript.

Model de dades : És un estàndard que defineix un conjunt de dades comuns que tant el LMS com el SCO han de conèixer. Ha estat desenvolupat per AICC CMI, Aviation Industry Computer-Based Training Committee, és una associació internacional de capacitació de professionals basada en tecnologia. El LMS monitoritza el SCO i aquest ha de utilitzar el conjunt de dades predefinit. Durant la comunicació i, mitjançant les funcions de l'API, els usuaris utilitzen aquest model de dades.

3.2.4 Comunicació

A continuació es detalla com es comuniquen els diferents mòduls de l'entorn de run-time basat en SCORM, un cop ja afegida l'activitat en el curs. figura 20 es resumeixen aquestes crides de comunicació entre les diferents entitats implicades.

1. Inici de la seqüència: trobant l'API Primerament, des de Moodle s'executa la activitat SCORM que s'ha creat. Aquesta llençarà un SCO al navegador del client amb els assets necessaris i una instància de l'API.

Exemple de comunicació diagrama de seqüències runtime service

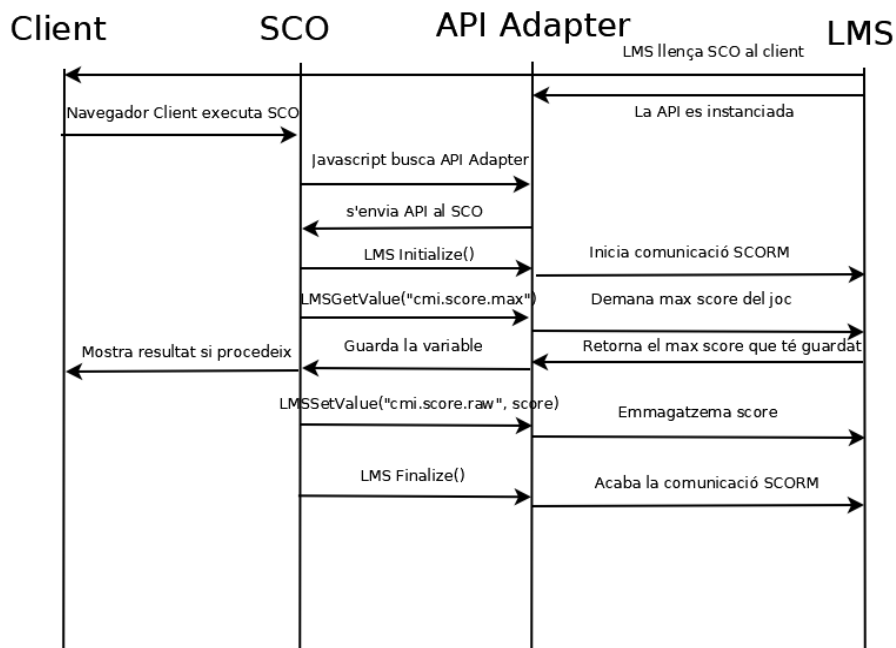


Figura 20: Comunicació LMS - SCO

Quan el SCO es carrega al client, aquest carrega els fitxers necessaris mitjançant el script que inicia la cerca de l'API Adapter. S'intenta fer aquesta cerca de l'API Adapter des de la finestra pare que, en cas de no trobar-se, es propaga la cerca en la cadena de finestres pare fins a la finestra arrel. Si tot i així no es troba l'API Adapter, es busca a la finestra oberta. En el cas que no es trobi la API en aquesta seqüència, es llança un error de fallida de la xarxa amb el servidor.

2. Triggers de comunicació La comunicació entre el LMS i el SCO ha de ser específicament entre dos flags o triggers. Per iniciar la comunicació es crida el trigger *LMSInitialize* i per finalitzar-la cal cridar el trigger *LMSFinish*. Tota comunicació del SCO que vagi fora d'aquests dos flags serà ignorat pel runtime service. *LMSInitialize* pot ser cridat una altra vegada pel SCO, o no, tot depenent de la implementació que s'hagi programat en el client.

Cal tenir en compte que l'usuari pot tancar la finestra de navegació i pot perdre's la informació del navegador. Per evitar que això passi cal cridar als mètodes de JavaScript *onunload* o *onbeforeunload*. Cal comentar que el segon mètode és millor que el primer ja que en el segon es disposen encara de les dades i, per tant, es poden fer les crides pertinents als mètodes de la API per salvar les dades.

3. Comunicació amb la API L'API que proporciona el LMS té unes funcions predefinides que proporciona la comunicació entre el SCO i el LMS, i la implementació d'aquesta està oculta al SCO. En el apartat anterior s'ha comentat un tipus de

funcions, que obre i tanquen, la comunicació entre LMS i SCO, aquestes funcions s'anomenen **funcions d'estat** i són les dues mencionades abans *LMSInitialize* i *LMSFinish*.

Després s'ofereixen tres **funcions que gestionen els errors** dels mètodes, que tot i que en aquest projecte no s'han utilitzar, estan implementades en el Javascript, *LMSGetLastError()*, *LMSGetErrorString()* i *LMSGetDiagnostic()*

Per acabar, es disposa de les **funcions de transferència**, que són funcions que parametrizen els valors i comuniquen dades rellevants entre LMS i SCO, tals com guardar una puntuació o el temps que ha tardat en solucionar un quizz. Aquestes funcions són les següents: *LMSGetValue()*, *LMSSetValue()* i *LMSCommit()*.

Com ja s'ha explicat anteriorment, l'API de Moodle s'instancia en el navegador per ser poder ser accessible des del Javascript i poder utilitzar les seves funcions i el model de dades. Per fer possible això, s'ha de programar una funcionalitat pròpia en el Javascript que accedirà aquesta API Adapter. Per a la nostra solució, s'han implantat dos tipus d'implementacions de l'API, una per JavaScript i una altra per Unity. Aquestes funcionalitats s'han anomenat API Javascript i API Unity.

3.3 API Javascript

En qualsevol cas la connexió que mostra la figura 21 comença al fitxer **imsmanifest.xml**. Aquest fitxer dóna informació important al LMS de SCORM, tals com tipus de diagrama, versió, títol, identificació o fitxers que ha de seguir el LMS de execució. A continuació, el fitxer *imsmanifest.xml* buscarà l'arxiu principal que executarà, que en aquest cas és l'**index.html**. Per tant, en el LMS s'obrirà un arxiu *index.html* en una finestra dins de l'activitat SCORM, aquesta mostrarà el contingut del arxiu *html* i farà les crides pertinents de Javascript per donar la funcionalitat a aquest *html*.

A part de cridar els codis de Javascript que realitzen la funcionalitat dels *.html*, també es començaran fent les crides de les funcions SCORM per agafar l'API Adapter instanciada al navegador.

Per tant, des del *.html* o un dels codis de Javascript es cridarà el *Initialize* i aquest cridarà la funció *getApi* i *buscarApi* fins trobar el API Adapter. A partir d'aquí el programa principal treballarà per una part, i les funcions SCORM's enviaran i rebran dades tal i com estiguin implementades al programa principal. S'acaba l'execució quan es crida la funció *Finish* on tancarà comunicació amb el LMS, o es tanqui inesperadament l'activitat SCORM. Per aquests casos, caldrà tenir desenvolupades les funcions *onbeforeunload* i *unload* i guardar la informació ràpidament ja sigui usant *SetScore* o qualsevol altra funció per enviar les dades a la base de dades del LMS.

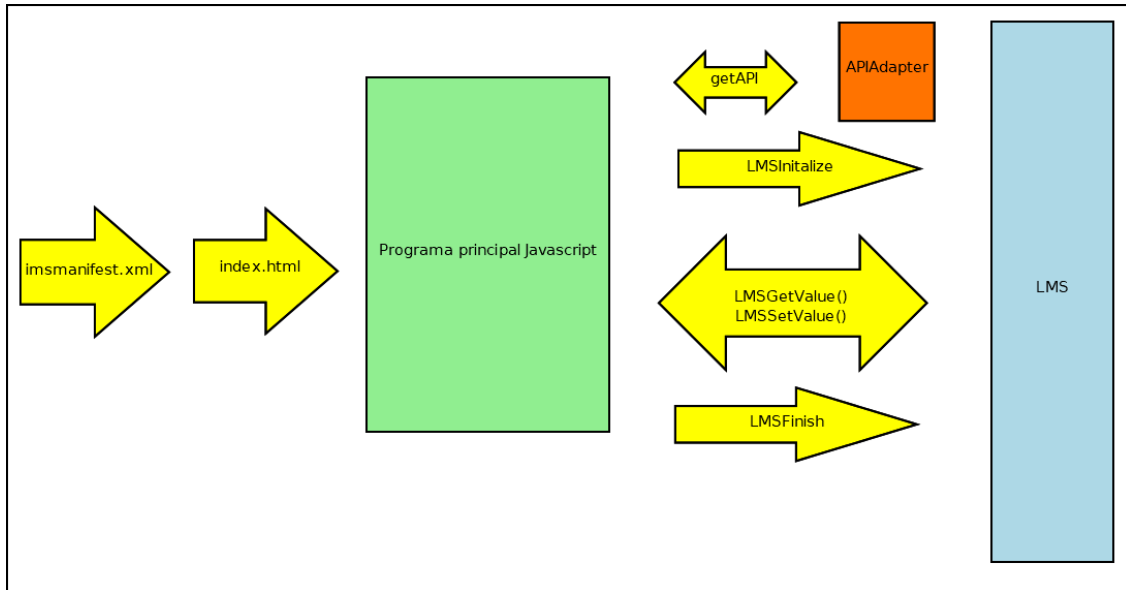


Figura 21: Comunicació Javascript

3.4 API Unity

La comunicació en Unity difereix una mica de com ho es fa en Javascript, però la idea general és la mateixa. Cal buscar l'API Adapter per comunicar-se amb el LMS. En aquest cas l'arquitectura per comunicar Unity amb Moodle ha de passar a través d'un Javascript de nou. En aquest cas, però, hi ha un problema afegit: les últimes versions de Unity ja no suporten el Javascript com a llenguatge de desenvolupament, així que s'ha optat per l'ús dels fitxers `.jslib` per a fer la comunicació amb el LMS. Aquest fitxers són suportades per Unity com a llibreries que poden ser cridades des del codi del joc escrit en C Sharp. La llibreria SCORM que es disposa en el asset store de Unity⁴, té totes les funcionalitats necessàries per comunicar-se amb el LMS incloent el fitxer `.jslib`, a més d'un fitxer com `ScormPublishSettings.asset22` que serveix per configurar el fitxer `imsmanifest.xml`.

A més a més, a Unity es té el problem agregat que cal traduir els programes fets en el llenguatge C Sharp a un llenguatge comprensible pels navegadors. S'ha optat per a utilitzar l'opció que ofereix Unity per a muntar el codi en WebGL en html5. Aquesta opció en el desplegament del projecte Unity genera uns fitxers en extensió `.unityweb` en hexadecimal, a més a més d'un fitxer en `.json` i un altre en Javascript, aquest últim és el fitxer que es cridarà des del `index.html` per iniciar les funcions Unity i SCORM.

Una vegada desplegat el codi, l'execució comença amb el `imsmanifest.xml` i després al `index.html`, tal i com es feia en l'API de Javascript descrita en l'apartat anterior. Arribat al `index.html` es fa una crida Javascript `UnityLoader.instantiate(..)` que carrega el fitxer generat amb WebGL. A partir d'aquí s'executa tot el codi C Sharp de Unity. Primerament, es cridarà el mètode `start()` de la classe principal

⁴<https://assetstore.unity.com/packages/tools/integration/scorm-api-53523>

del joc. La API de Unity que s'utilitza per anar a buscar el API Adapter utilitza un fitxer C Sharp que s'anomena *ScormService.cs* i una classe del format de llibreria de Javascript, anomenada *Scorm.jslib*. La classe *ScormService* s'instancia en el mètode start, que en primer loc, executa el mètode *Initialize*, però que fa realment aquesta classe per comunicar-se amb el Javascript? La resposta són els Imports que té a la capçalera de la classe per realitzar les crides del fitxer *Scorm.jslib*, on té la API de Javascript, encapsulat en el mètode *mergeIntro*. Aquest fluxe es pot veure a la figura 23.

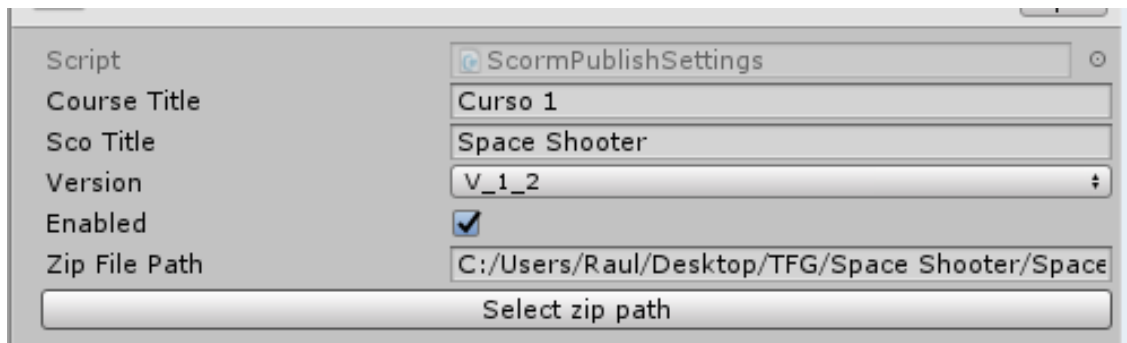


Figura 22: Configuració paràmetres imsmanifest.xml per Unity

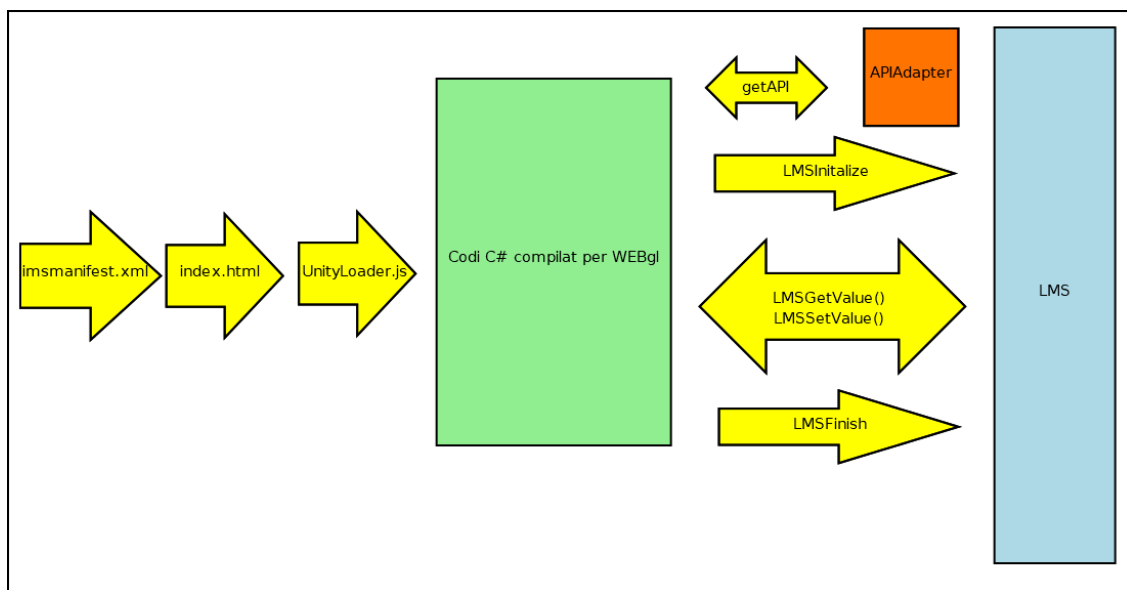


Figura 23: Arquitectura Unity-Scorm

4 Resultats i Simulacions

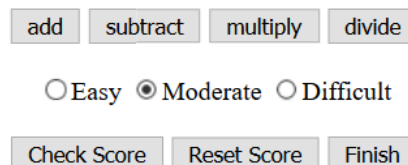
A continuació es presenten els tres serious games que s'han desenvolupat i adaptat per aquest projecte. S'ha partit d'una implementació bàsica ja donada de cada joc. En tots ells, s'han realitzat diverses modificacions per adaptar la llibreria SCORM en cadascun dels seus llenguatges de programació. En alguns dels jocs s'han fet diverses modificacions per passar d'un joc sense enfoc pedagògic a un que tingués un enfoc per activitats estudiantils.

4.1 Math Quiz

4.1.1 Explicació joc

Math Quiz[6] és un joc programat en Javascript i sense interfície gràfica o GUI. L'objectiu del joc és aconseguir la màxima puntuació encertant tots els càlculs que proposen. El joc té un número infinit de càlculs, sempre que es premi al botó de següent partida, i fins que l'usuari es cansi de jugar, ja que pot deixar el joc quan vulgui. En acabar el joc, es farà una mitja d'encerts i errors, aquest percentatge serà la puntuació final del usuari.

L'interfície principal del joc és un formulari de la figura 24 en html que disposa de set botons i tres radio-buttons. Per jugar, l'usuari haurà d'escollir la dificultat del càlcul, per després escollir entre quin tipus de càlcul voldrà realitzar (suma, resta, multiplicació i divisió). A continuació sortirà un quadre de text on el joc pregunta per la resposta correcta d'un càlcul com el de la figura 25. Si la resposta és correcta sortirà un quadre de text enunciant l'enhorabona i fent un promig d'encerts/fallades, figura 26, si es fallés sortiria un text de fallada i també fent el promig d'encerts/fallades. Aquesta serà la rutina que anirà experimentant l'usuari al jugar, però a més a més el joc disposa de tres opcions més. Una opció per reiniciar la puntuació, un altre per realitzar el test d'encerts i errors, i finalment un últim botó per acabar el joc.



The image shows a web form for a math quiz. It contains four buttons for operations: 'add', 'subtract', 'multiply', and 'divide'. Below these are three radio buttons for difficulty levels: 'Easy', 'Moderate' (which is selected), and 'Difficult'. At the bottom are three buttons: 'Check Score', 'Reset Score', and 'Finish'.

Figura 24: formulari math quiz

El format que disposa aquest joc ja està enfocat com a serious game. Tot i així no té una interfàç atractiva d'usuari, ja que és un simple formulari, però amb això en tenim suficient per realitzar les primes proves amb la llibreria SCORM.

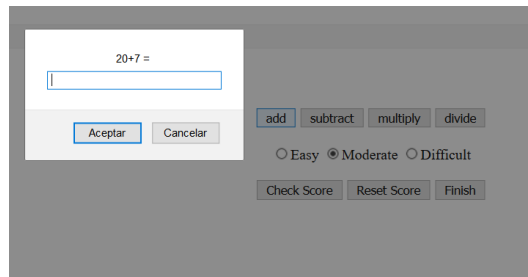


Figura 25: pregunta math quiz

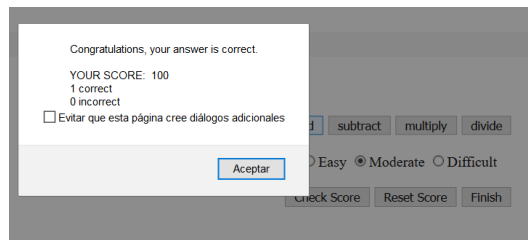


Figura 26: càlcul correct math quiz

4.2 Math Quiz a Moodle

La intenció d'integrar aquest serious game és comprovar que es pot comunicar mitjançant SCORM amb Moodle. El que es vol aconseguir és comprovar que en un mateix intent del joc es pot recuperar l'intent anterior i enviar a Moodle la nova puntuació aconseguida.

Per tant, l'alumne començarà a jugar directament sense que el Professor tingui que parametritzar cap dada des de la seva vista. L'alumne accedirà a l'activitat SCORM on podrà visualitzar el resultat de l'últim intent, figura 27 i en prémer el botó **Entrar**, s'accedirà al joc.

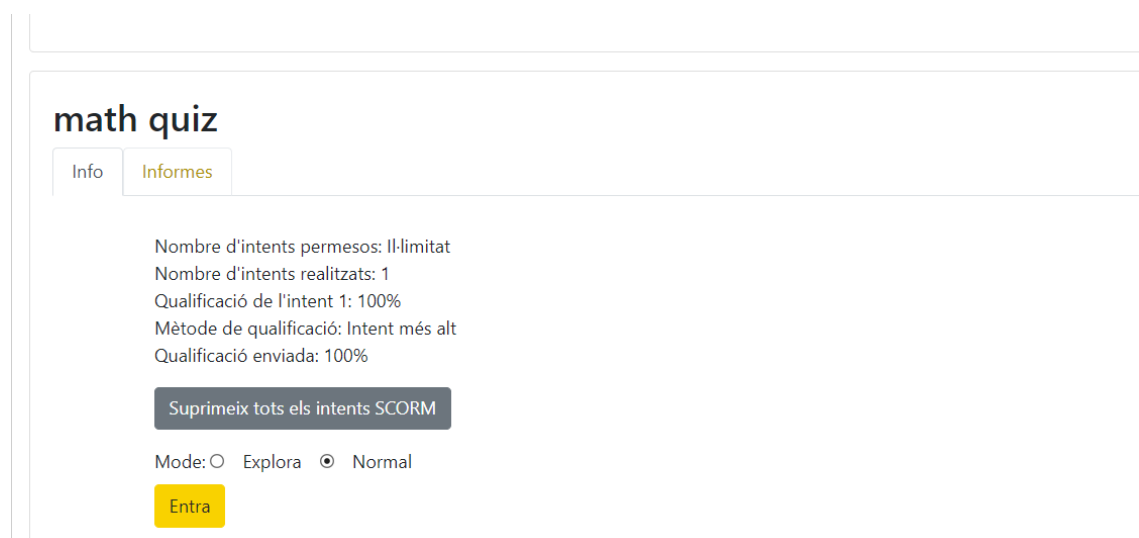


Figura 27: Abans de començar l'intent math quiz

Després d'haver jugat al joc i iniciat un nou intent, s'observa que en la finestra d'intents, figura 29, s'ha guardat l'intent anterior i a més a més s'obté també aquesta puntuació, mostrant la finestra el resultat en executar el joc, figura 28.

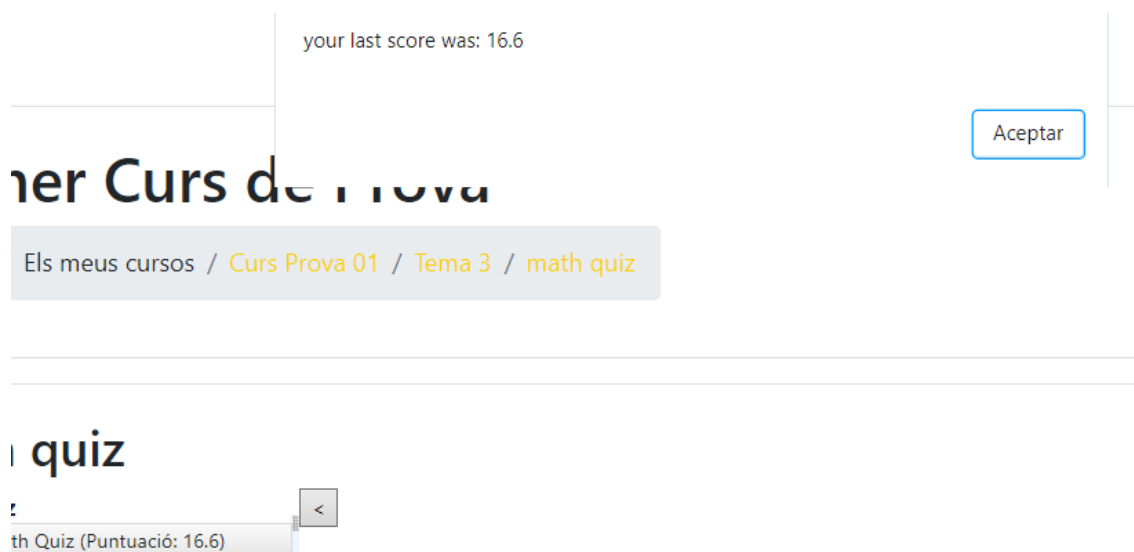


Figura 28: Resultat anterior des del joc

4.3 Breakout

4.3.1 Explicació joc

Breakout o arkanoid [7] [8] és un joc-tutorial desenvolupat en Javascript utilitzant el framework *the Javascript Phaser* amb una interfície gràfica.

L'objectiu d'aquest joc és desfer-se de tots els blocs que hi han en la zona d'amunt de la pantalla amb una bola que s'ha de fer rebotar amb una paleta. Aquesta paleta es pot moure amb el ratolí d'un cantó a un altre, evitant que la pilota desaparegui per el cantó de sota de la pantalla. L'esfera va rebotant pels costats i per damunt de la pantalla, a la vegada que va col·lisionant amb les totxanes. A més a més, depenent amb quina part de la paleta li dóna a la bola, aquesta agafarà un efecte diferent. Els blocs van desapareixent a mesura que la pilota col·lisiona amb aquests i per cada colisió es guanya una puntuació. El jugador té inicialment tres vides i en perd una cada vegada que la pilota cau fora de la pantalla. El jugador pot seguir jugant fins a perdre les tres vides, gameplay a la figura 30.

Aquesta es la versió inicial del joc, que s'ha modificat per a transformar-se en una activitat educativa (o serious game) i integrar-lo com a possible activitat de Moodle.

math quiz

Info

Informes

Nombre d'intents permesos: Il·limitat

Nombre d'intents realitzats: 1

Qualificació de l'intent 1: 17%

Mètode de qualificació: Intent més alt

Qualificació enviada: 17%

Suprimeix tots els intents SCORM

Mode: ☐ Explora ☒ Normal

Entra

Figura 29: Resultat anterior des de la pantalla d'intents

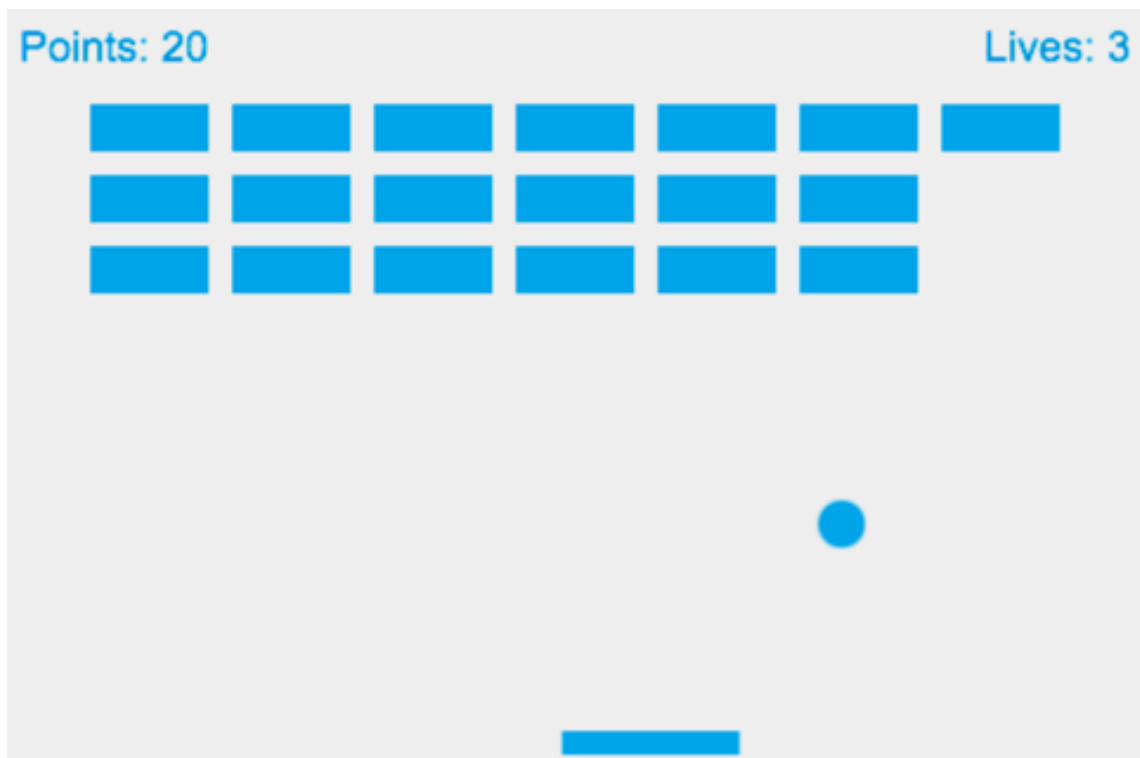


Figura 30: Breakout joc base

4.3.2 Serious Game Breakout

Per adaptar el joc breakout com a serious game s'ha pensat en l'enfocament que es trobarà una vegada es doni d'alta a Moodle. És a dir, es vol usar un mòdul scorm que s'encarregui de monitoritzar serious game i sigui un estàndard alhora de configurar-los. Per tant, un enfocament senzill per aquest tipus de joc és transformar-lo en una calculadora mentre es juga a temps real.

El primer canvi realitzat en el joc és el canvi de l'objectiu. En principi el jugador veurà un càlcul al cap d'amunt de la pantalla. Aquest càlcul s'ha de resoldre, i les solucions estan visibles en els blocs. Cada bloc té un número diferent i el jugador caldrà que encerti el bloc que dona el resultat de l'operació plantejada.

Internament els números de blocs es generen aleatòriament exceptuant un parell, que depenen del número de blocs que queden en pantalla. El càlcul de números aleatoris i resultats correctes es calculen usant un algorisme que es basa en el número de blocs actuals que queden en pantalla i d'un mínim de resultats correctes que es forcen a sortir en els blocs, gameplay figura 31.

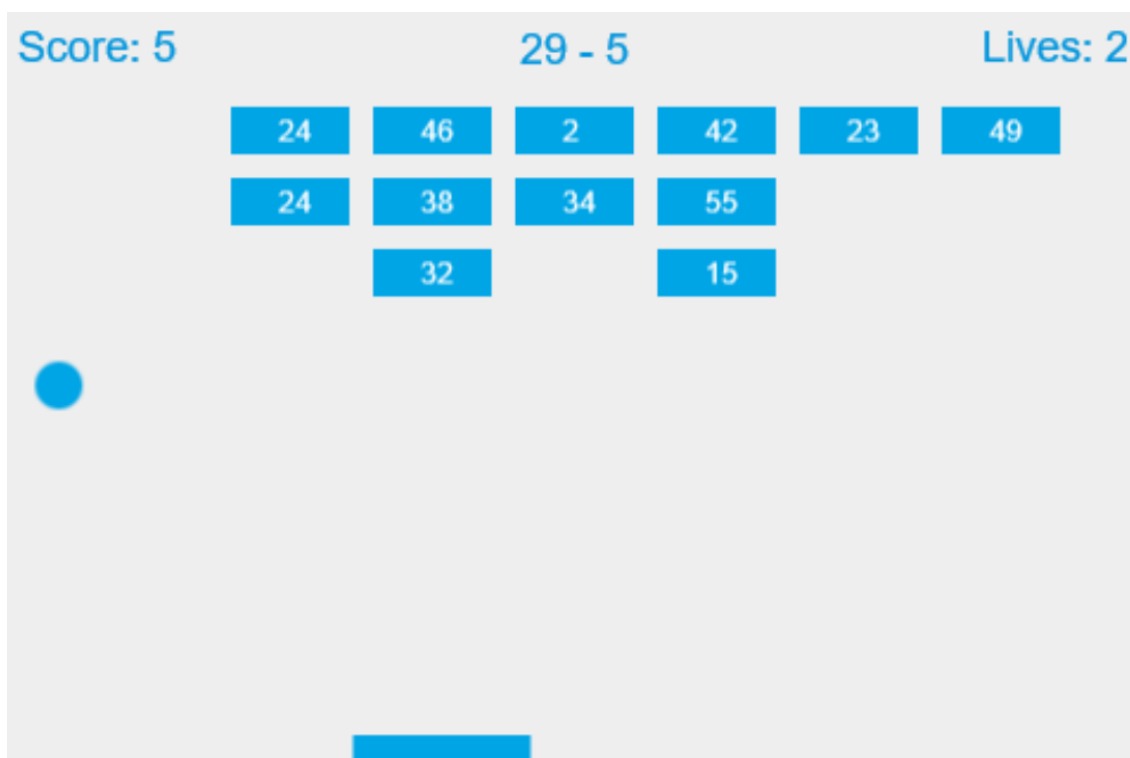


Figura 31: Serious game breakout

4.3.3 Serious Game Breakout amb Moodle

La integració a Moodle del breakout es configura amb uns paràmetres del joc des del formulari de la vista del professor, formulari en la figura 32. Aquests opcions que cal configurar són un número, un tipus de signe per realitzar el càlcul i un

nivell de dificultat. L'usuari professor podrà configurar aquestes opcions accedint a l'activitat del mòdul SCORM que es guardaran a la taula Moodle.

Curso1

Àrea personal / Mis cursos / curso1 / Tema 2 / breakout

breakout

Info Informes

▼ Valors Scorm

Número

9

Signe

☐ Suma ☐ Resta ☒ Multiplicació ☐ Divisió

Dificultat

☒ Fàcil ☐ Normal ☐ Difícil

Guardar cambios

Cancelar

Figura 32: Configuració del joc al formulari des de la vista del professor

En guardar el resultat en aquesta vista, l'alumne pot accedir al joc amb els paràmetres preestablerts pel professor. El primer que es pot observar en entrar a l'activitat de la figura 33 és el número d'intents que es porten fets, la puntuació màxima que ha obtingut i un botó que servirà per començar el joc.

Internament, en el moment d'executar-se el joc com a la figura 34, el primer que es farà és instanciar el SCO i realitzar la crida a les funcions Javascript per recollir els paràmetres scorm que li està enviant Moodle. En rebre els paràmetres, el joc ja pot començar, en donar al botó **start** del joc, comença a moure's la pilota. Quan es col·lisiona amb un dels blocs, es restaran cinc punts si la resposta al càlcul es incorrecte i es sumaran 20 punts si el càlcul es correcte. En finalitzar el joc, s'envia la puntuació final de l'usuari via la comunicació de scorm amb Moodle, i es poden visualitzar als informes del professor.

4.3.4 Explicació joc

Space Shooter [9] és un joc-tutorial desenvolupat en Unity amb GUI i programat en C Sharp. L'objectiu d'aquest joc es arribar a eliminar el màxim d'enemics i fer la màxima puntuació possible. Una vegada el joc s'acaba, es pot tornar a reintentar tantes vegades com es desitgi. El tutorial base del joc arriba només a mostrar una part del joc. El joc que es mostrarà en aquest projecte s'ha arribat a realitzar la part avançada del tutorial també, que inclou més asteroides, enemics o donar moviment al fons de pantalla.

La interfície principal del joc és un fons fosc de pantalla, emulant el buit univers. L'usuari ha de manipular la nau que es pot veure a sota de la pantalla quan inicia el joc, podent moure-la de dreta a esquerra i amunt i avall fins a uns límits establerts dins del joc. Des d'amunt de tot la pantalla aniran caient asteroides i enemics com en la figura 35 que l'usuari haurà d'esquivar ja sigui movent-se o disparant amb el botó esquerra del ratolí. A mesura que l'usuari vagi derrotant adversaris, s'afegeix una puntuació de 10 punts al usuari, veure la figura 36. Una vegada l'usuari col·lisiona amb un dels enemics o asteroides, el joc s'acaba i surt una pantalla de Game Over

breakout

prova n

Número de intentos permitidos: Sin límite

Número de intentos realizados: 19

Calificación del intento 1: 20%

Calificación del intento 2: 0%

Calificación del intento 3: 0%

Calificación del intento 4: 0%

Calificación del intento 5: 0%

Calificación del intento 6: 45%

Calificación del intento 7: 0%

Calificación del intento 8: 0%

Calificación del intento 9: 0%

Calificación del intento 10: 0%

Calificación del intento 11: 20%

Calificación del intento 12: 0%

Calificación del intento 13: 0%

Calificación del intento 14: 0%

Calificación del intento 15: 0%

Calificación del intento 16: 0%

Calificación del intento 17: 0%

Calificación del intento 18: 0%

Calificación del intento 19: 0%

Método de calificación: Intento más alto

Calificación informada: 45%

Moda: ☐ Vista previa ☒ Normal

Entrar

Figura 33: Configuració del joc al formulari des de la vista del professor

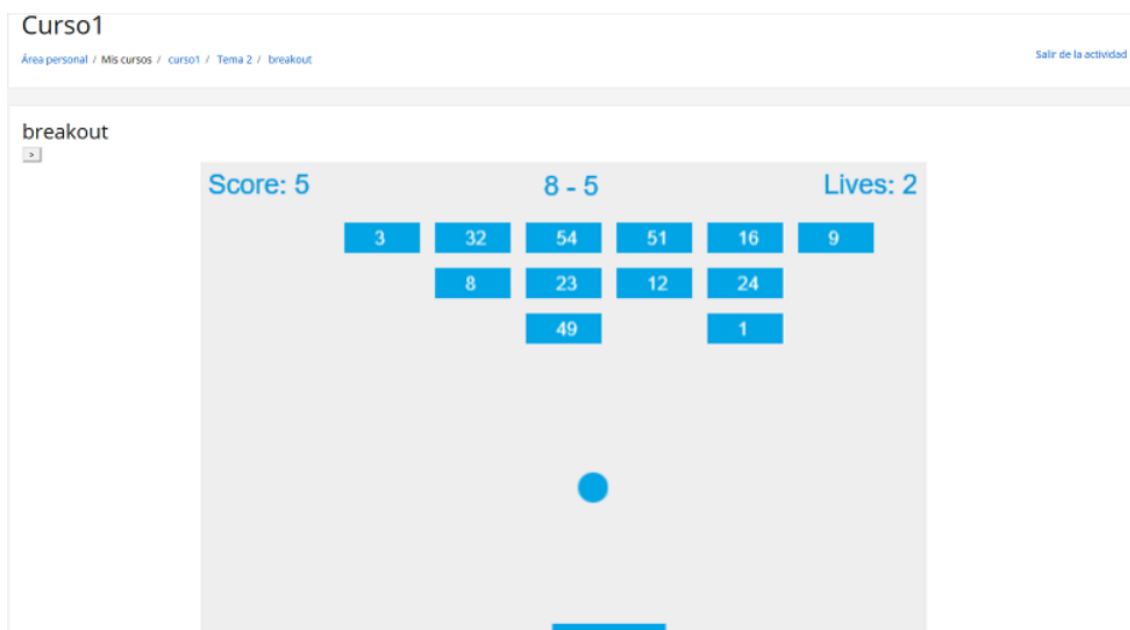


Figura 34: Visualització del joc en l'activitat de Moodle

com en la figura 37 enmig de la pantalla. Al cap d'uns segons el joc donarà la opció de reinici mitjançant el botó R.

Fins aquí s'ha vist un joc base desenvolupat a partir d'un tutorial de Unity. Es un joc enfocat a l'entreteniment i no està enfocat gens a l'aprenentatge. En el següent apartat s'expliquen les modificacions que s'han introduït per a transformar-lo en un mini-serious game.

4.3.5 Serious Game Space Shooter

L'adaptació del joc com a serious game, que es pot veure en la figura 38), es materialitza en un joc d'encertar resultats a partir d'un càlcul matemàtic i la recompensa d'aquests en forma de score positiu, i penalització amb score negatiu.

Per tant, els primers canvis que es veurà en el joc serà un petit text al cap d'amunt del joc on hi haurà una operació a resoldre. El següent canvi que es veurà són els asteroides i enemics, que tenen inserits un número, en el cas dels enemics ben clars. En els asteroides, com van rotant sobre si mateixos, cal que també rotin els números associats a ells. Per últim, es mostra el feedback de resposta en encertar o errar en el objectiu que s'ha de respondre correctament (encertar donarà 20 punts, mentre que errar restarà 5 punts).

En principi, el joc està pensat per a configurar-se a partir d'unes dades donades pel professor. Les dades que es poden configurar són: número, signe i nivell, i són utilitzades per un algoritme de creació de números aleatoris dins d'un rang o per realitzar el càlcul. Aquest algoritme es realitza dins del joc.

En la següent secció s'explica com aquestes dades arribaran a establir-se com a paràmetres del joc a partir de la comunicació amb el LMS.

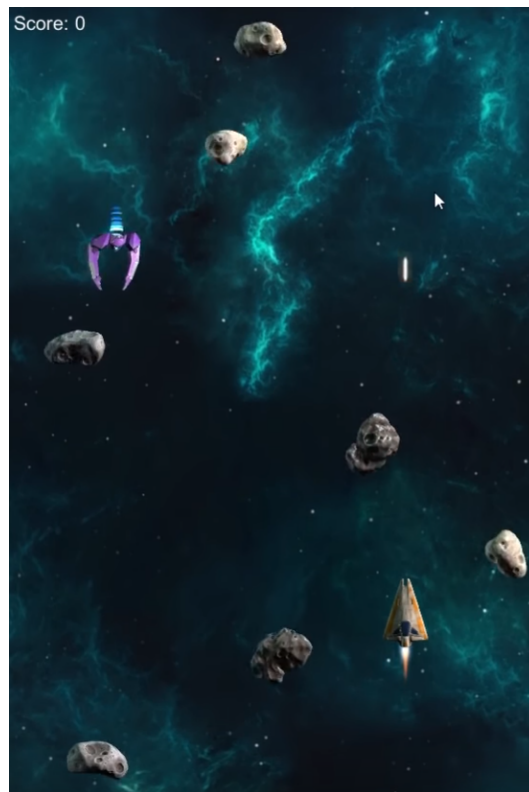


Figura 35: jugant space shooter

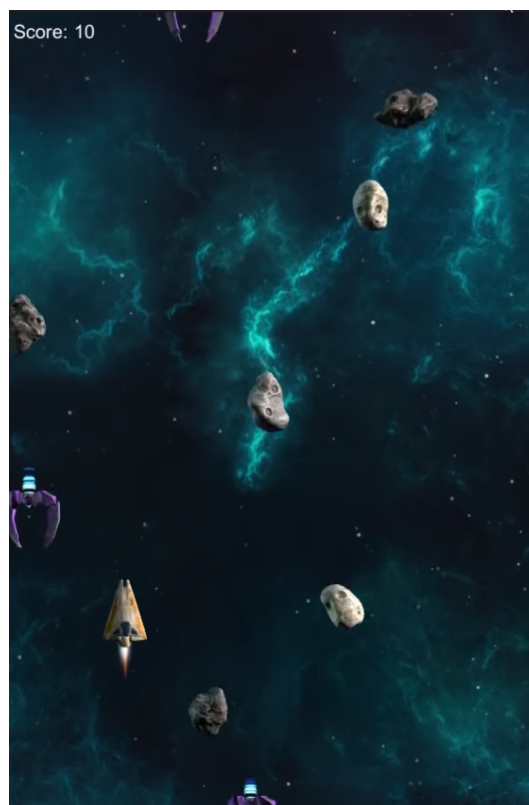


Figura 36: jugant space shooter

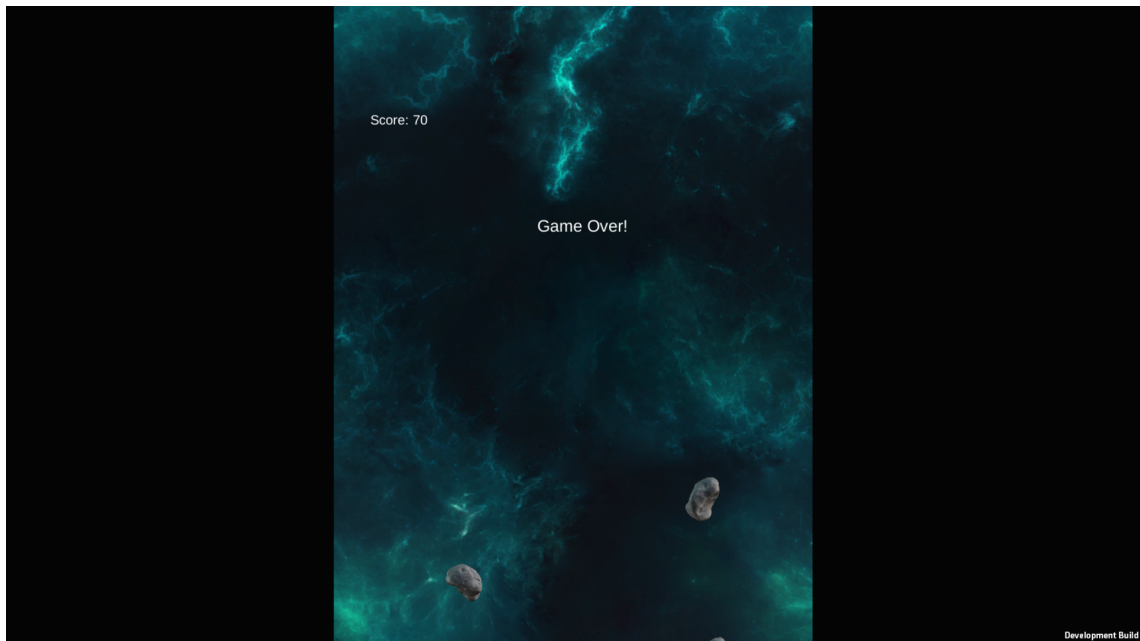


Figura 37: Joc acabat

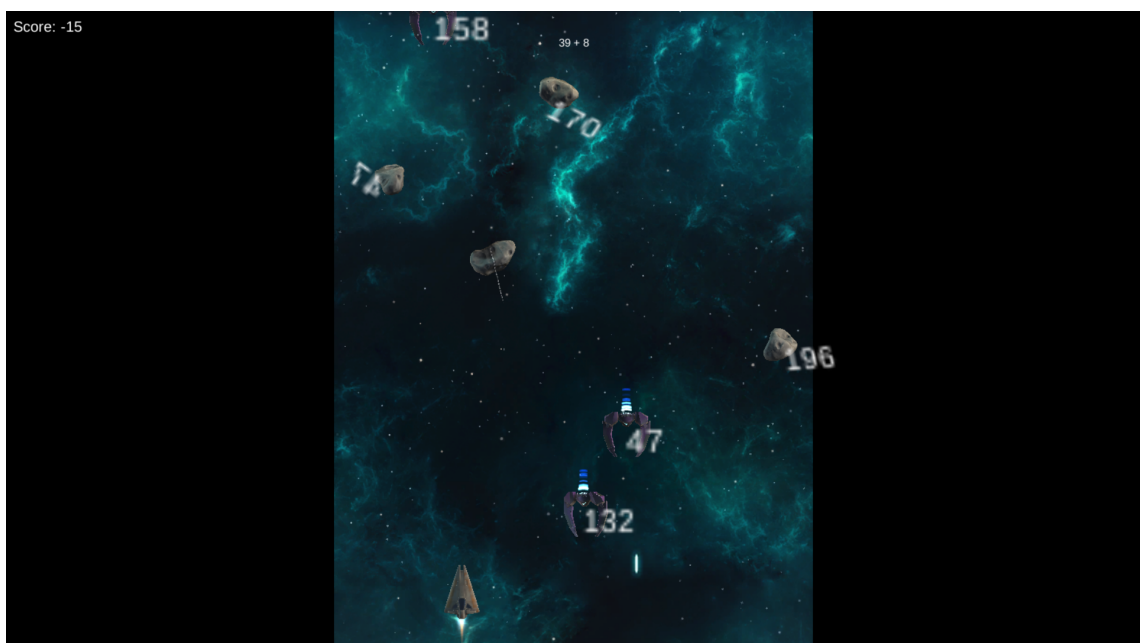


Figura 38: serious game space shooter

4.3.6 Serious Game Spacer Shooter amb Moodle

La connexió amb Moodle del serious game comença amb la parametrització dels valors que s'han d'establir des d'un rol de professor. Un usuari amb rol de professor i matriculat en el curs en concret que està pujat el serious game, ha d'establir les propietats mencionades en el punt anterior (número, signe i nivell). Una vegada entra el mòdul SCORM, el professor es trobarà un formulari com el de la figura 39, en el que haurà de configurar els paràmetres. Una vegada s'han configurat, aquests paràmetres es guarden en la taula *mdl_scorm_parameters_data* interna de Moodle.

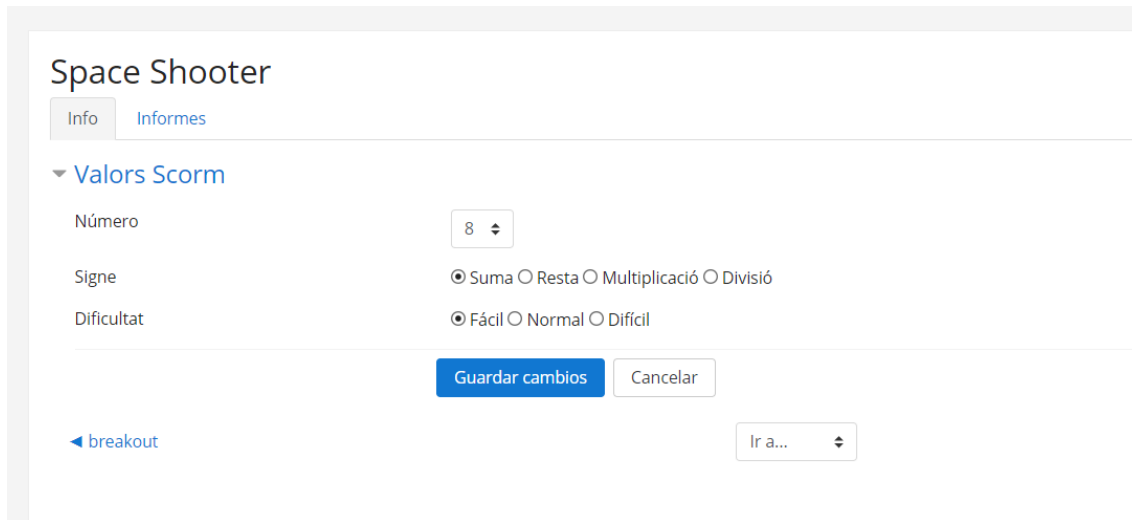


Figura 39: Formulari de configuració des de la vista del Professor

D'altra banda, l'alumne podrà accedir al scorm del Space Shooter i en la seva finestra principal trobarà un resum dels intents anteriors, com els de la figura 40, on l'últim és l'intent actual, amb les puntuacions i un botó per entrar al joc. Una vegada pren el botó **entrar**, l'alumne troba la finestra de càrrega de Unity per després iniciar el joc. Quan comenci a jugar pot maximitzar la finestra com es mostra a la figura 41 de Unity al començar a jugar.

Com s'ha pogut observar, en canviar la configuració des del professor, varien els paràmetres que rep l'alumne al joc, a partir de la connexió amb SCORM. Per aquests tipus d'activitat, l'alumne podrà provar tantes vegades com vulgui el joc fins que el professor tanqui l'activitat. En quant al professor, pot veure el feedback de tots els intents que l'alumne ha realitzat en la pestanya d'Informes del mòdul, com es pot veure en la figura 42.

Space Shooter

Space Shooter

Número de intentos permitidos: Sin límite

Número de intentos realizados: 22

Calificación del intento 1: 55%

Calificación del intento 2: 0%

Calificación del intento 3: 0%

Calificación del intento 4: 0%

Calificación del intento 5: 20%

Calificación del intento 6: 35%

Calificación del intento 7: 15%

Calificación del intento 8: 80%

Calificación del intento 9: 100%

Calificación del intento 10: 35%

Calificación del intento 11: 0%

Calificación del intento 12: 0%

Calificación del intento 13: 20%

Calificación del intento 14: 0%

Calificación del intento 15: 0%

Calificación del intento 16: 0%

Calificación del intento 17: 75%

Calificación del intento 18: 0%

Calificación del intento 19: 0%

Calificación del intento 20: 40%

Calificación del intento 21: 0%

Calificación del intento 22: 0%

Método de calificación: Intento más alto

Calificación informada: 100%

Moda: ☐ Vista previa ☒ Normal

Entrar

Figura 40: Finestra abans de començar el joc, vista d'alumne

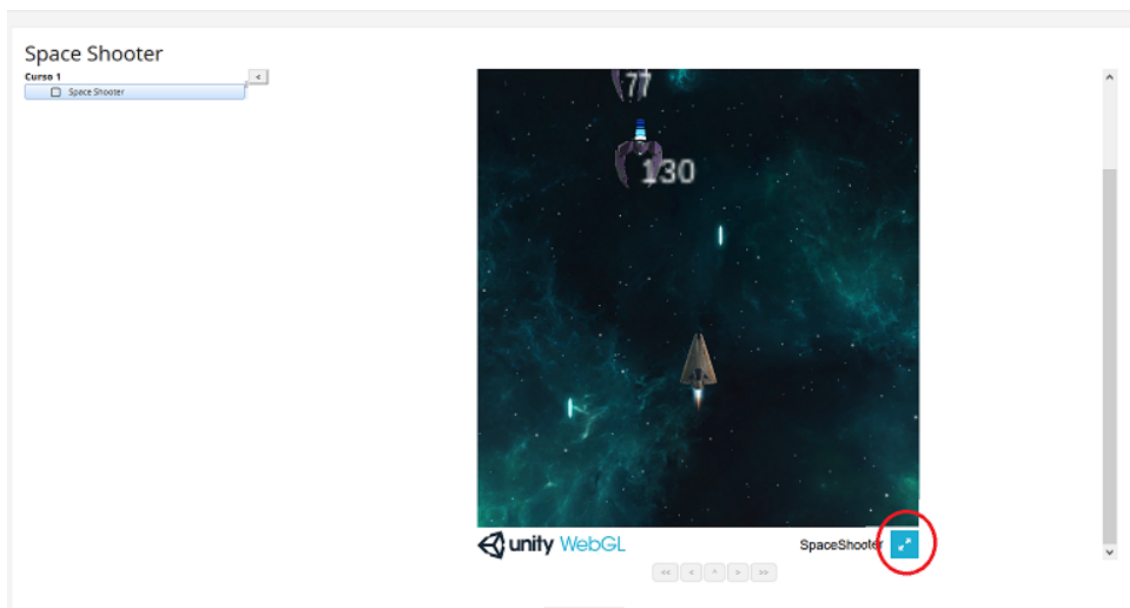


Figura 41: Pulsar el següent botó per gaudir d'una bona experiència

	Nombre / Apellido(s)	Dirección de correo	Intento	Comenzado en	Último acceso en	Puntuación
	Estudiante Alumne	estudiante@gmail.com	-	-	-	-
<input type="checkbox"/>			11	jueves, 20 de junio de 2019, 17:39	jueves, 20 de junio de 2019, 17:39	0
<input type="checkbox"/>			12	jueves, 20 de junio de 2019, 21:45	jueves, 20 de junio de 2019, 21:45	0
			-	-	-	-
			-	-	-	-
			-	-	-	-
<input type="checkbox"/>			17	jueves, 20 de junio de 2019, 22:43	jueves, 20 de junio de 2019, 22:44	75
			-	-	-	-
<input type="checkbox"/>			2	miércoles, 19 de junio de 2019, 18:15	miércoles, 19 de junio de 2019, 18:16	0
			-	-	-	-
			-	-	-	-
<input type="checkbox"/>			22	martes, 25 de junio de 2019, 02:47	martes, 25 de junio de 2019, 03:00	0
			-	-	-	-
			-	-	-	-
<input type="checkbox"/>			25	martes, 25 de junio de 2019, 03:10	martes, 25 de junio de 2019, 03:11	80
			-	-	-	-

Figura 42: Feedback de score al Professor

5 Conclusions i feina futura

L'objectiu d'aquest projecte era el de definir un procés per tal d'incloure serious games en una plataforma LMS a través d'una API de comunicació que pogués transmetre i rebre dades per diferents plataformes de jocs. En un principi, es va planejar la inclusió de diferents serious games: un joc en Javascript sense GUI, un altre joc en Javascript amb GUI i finalment un tercer joc en Unity amb GUI. Alhora d'incloure els jocs, un requisit important era que en el LMS hi convivien dos tipus de usuaris diferents, amb diferents rols, i que cadascun tindria una tasca diferent alhora d'interactuar amb els serious games. El rol de professor tenia la tasca de configurar uns paràmetres en un formulari, per després guardar-los en una base de dades, perquè l'alumne jugués amb aquesta configuració que s'havia realitzat. Alhora de jugar el serious game, l'alumne tenia un doble repte, primerament havia de ser hàbil per avançar en el joc i també hàbil per fer càlculs matemàtics al mateix temps que juga.

Les contribucions que s'han realitzat en aquest projecte són les següents:

- Disseny, proves i implementació de l'API SCORM a Javascript.
- Implementació de l'API SCORM a Javascript en un serious game sense GUI.
- Adaptació d'un joc de Javascript amb GUI per convertir-lo en serious game i implementació de l'API SCORM en aquest.
- Adaptació d'un joc de Unity amb GUI per convertir-lo en serious game i implementació de l'API SCORM en aquest.
- Modificació del mòdul SCORM de Moodle, adaptar-ho a les necessitats del projecte i usuaris que l'utilitzaran.
- Integració de tres serious games a Moodle amb comunicació.

Després d'haver provat els serious games en la plataforma de Moodle, es pot afirmar que s'han assolit cadascun dels objectius. S'han analitzat diferents LMS i s'ha seleccionat un LMS que complia els requeriments funcionals partidada. A més a més s'ha escollit un protocol de comunicació que s'integra perfectament al LMS. S'ha realitzat l'anàlisi de l'API de comunicació per veure l'abast dels mètodes i el model de dades per després fer el disseny en cadascun dels llenguatges de desenvolupament propis dels serious games. S'han implementat i adaptat uns jocs base per a que fossin educatius i per a poder-los connectar amb Moodle. A partir d'aquestes proves realitzades s'han integrat els tres serious games amb el LMS, i s'han realitzat diverses proves de funcionament.

Aquest projecte pot continuar de diverses formes, tot depèn de quin camí es vol seguir. Com es comentarà en el apèndix, una manera d'ampliar el projecte seria monitoritzar més dades. Un bon exemple seria rebre les vides del breakout una vegada es s'han parametritzat a Moodle. Una altra forma d'ampliar el projecte seria fent un estudi més exhaustiu del funcionament del mòdul SCORM a Moodle, de manera

que no depengui de la implementació del LMS en aquest mòdul i es pogués enviar i rebre dades de una manera més lliure. Un altre exemple de continuació del projecte podria ser fer diferents instàncies del mòdul SCORM de manera que es pugui posar un context diferent al matemàtic, o al que s'ha posat en aquest projecte que són les diverses operacions, i focalitzar-lo en un més en un aprenentatge lingüístic, com el posar la paraula que falta a una frase. Un últim exemple d'ampliació de projecte podria ser la gamificació de diversos mòduls propis de Moodle, per animar a realitzar tot tipus d'activitats a Moodle.

A Manual tècnic

A.1 Instal·lació: Requeriments mínims i passos a seguir

La instal·lació de la infraestructura requereix d'uns requeriments no funcionals mínims (hardware - software) perquè sigui possible aquest desenvolupament.

En quant a requeriments hardware amb un ordinador dels que es fan servir avui dia ja ni hauria prou, tal com: processador Core i5, 4 o 8 GB de RAM, disc dur com es desitgi amb una capacitat mínima d'uns 50 GB o si es vol comprar un disc dur SSD a petició del consumidor, amb els perifèrics necessaris pels intercanvis de dades (ratolí, pantalla, teclat) i una connexió Internet de fibra òptica estàndard.

En quant a requeriments software és necessari un sistema operatiu Windows10 amb el software següent instal·lat: un descompressor, el paquet XAMPP (Apache, MariaDB, php), aquest paquet requereix Microsoft Visual C++ 2017 Redistributable.

La versió de Moodle que s'ha instal·lat és la 3.4 la versió portable de Windows que inclou XAMPP. Caldría configurar aquest paquet XAMPP per diversos temes de seguretat que el fan vulnerable a diversos atacs. XAMPP ve per defecte amb l'usuari root sense contrasenya, caldría canviar-la on s'executa en /server/mysql/bin i amb aquest exemple de comanda: **mysqladmin -user=root password "123456"** es modificaria, a més a més caldría realitzar el canvi de contrasenya al fitxer config.php de Moodle en l'atribut **\$CFG->dbpass** . En principi .htaccess ja esta configurat per Moodle, però caldría revisar les seves directives de access. Des de la versió 2.0 de Moodle, ja té mètodes per evitar atacs tipus sql-injection usant el seu framework que ho evita.

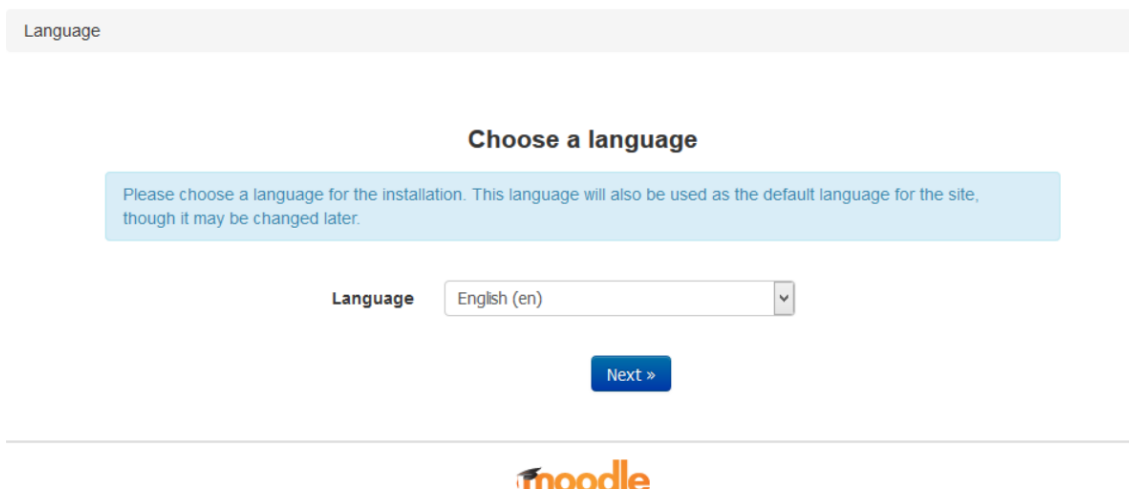
La instal·lació de Moodle és bastant senzilla i fàcil de seguir ⁵. Una vegada esta configurat XAMPP s'executa, això fa que el serveis de mariaDB i apache comencin a executar-se també. S'obre un navegador i s'escriu en aquest *localhost*, amb això hauria de sortir la pàgina d'instal·lació de Moodle com es pot veure a la figura 43. A partir d'aquí s'ha d'anar passant de pantalla amb els botons de següent, revisant sempre que les opcions són més o menys coherents i donant temps perquè s'instal·lin les bases de dades. En un moment determinat de la instal·lació demanarà un correu, i una contrasenya pel usuari administrador, cal recordar que una vegada instal·lat Moodle, l'usuari administrador és el que tindrà poders per afegir usuaris, i els diferents rols.

Una vegada la instal·lació s'ha completat, i s'han afegit com a mínim un curs, un usuari professor (editingteacher) i un alumne, es pot configurar el mòdul SCORM, pero abans de res, cal fer una configuració addicional i necessària.

Com a usuari administrador, cal anar a l'administració del lloc -> extensions -> vista general d'extensions i es busca el paquet SCORM per desinstal·lar-ho. Cal anar en compte, perquè el que es vol no és eliminar el mòdul totalment, sinó

⁵<https://moodle.org/file.php/11/moodlewindowsinstaller/MoodleWindowsInstaller.pdf>

Installation



The screenshot shows the Moodle installation language selection interface. At the top, there is a header bar with the word "Language". Below this, the title "Choose a language" is centered. A light blue box contains the instruction: "Please choose a language for the installation. This language will also be used as the default language for the site, though it may be changed later." Below the instruction, there is a label "Language" followed by a dropdown menu currently showing "English (en)". A blue "Next »" button is positioned below the dropdown. At the bottom of the page, the Moodle logo is displayed.

Figura 43: pàgina d'instal·lació de Moodle

desinstal·lar-lo per després instal·lar una taula necessària alhora de monitoritzar els jocs. Per tant, alhora de desinstal·lar-ho, avisarà si es vol eliminar completament i es pulsarà **Cancel·lar**, tal i com es veu a la figura 44.

Ara es copiarà la següent taula que es pot veure a la figura 45, en el fitxer `/mod/scorm/db/install.xml` del directori de Moodle, si pot ser entre dues taules ja implementades millor perquè no es posin en un lloc que no toca. Per acabar, s'ha d'actualitzar la finestra de Moodle on s'havia desinstal·lat el mòdul, que en actualitzar-la es podrà instal·lar novament.

A.2 Manual del desenvolupador: afegir un nou joc

Es pot arribar a modificar paràmetres de Moodle per ampliar o implementar noves funcionalitats de monitorització, però cal saber quins són els punts claus on cal afegir aquesta nova informació o implementació dins del codi font.

A.2.1 Moodle

Si es volgués afegir un nou paràmetre per enviar SCORM, al codi php de Moodle és on s'haurien de realitzar els majors canvis. Per començar, cal fer una menció de la taula que s'ha afegit al mòdul SCORM de Moodle `mdl_scorm_parameters_data` on s'hauria d'afegir un camp més.

En quant al codi php, el canvis serien en els següents fitxers: *mod_form_teacher.php* i *view.php*. El primer fitxer els canvis serien senzills, ja que primer fa una consulta a la base de dades per veure si hi han dades en la taula, i si n'hi han recupera aquestes dades per visualitzar-les en la vista del formulari. La següent part seria la declaració de la classe *values_scorm_mod_form* on s'haurien de declarar els input

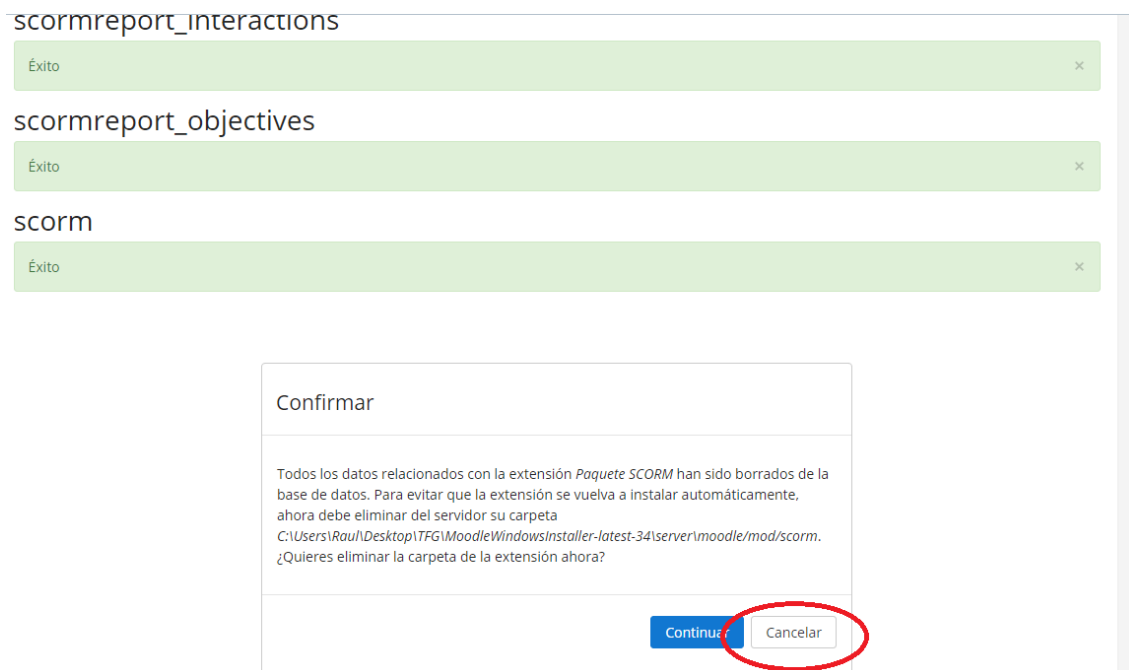


Figura 44: Instal·lació taula adicional al mòdul SCORM

```
<TABLE NAME="scorm_parameters_data" COMMENT="Contain variables to play javascript's games">
<FIELDS>
  <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true" SEQUENCE="true"/>
  <FIELD NAME="number" TYPE="int" LENGTH="10" NOTNULL="true" DEFAULT="0" SEQUENCE="false"/>
  <FIELD NAME="signe" TYPE="char" LENGTH="20" NOTNULL="true" SEQUENCE="false"/>
  <FIELD NAME="level" TYPE="int" LENGTH="10" NOTNULL="true" DEFAULT="0" SEQUENCE="false"/>
  <FIELD NAME="scorm" TYPE="int" LENGTH="10" NOTNULL="true" DEFAULT="0" SEQUENCE="false"/>
  <FIELD NAME="timecreated" TYPE="int" LENGTH="10" NOTNULL="true" DEFAULT="0" SEQUENCE="false" COMMENT="time
this session was created"/>
  <FIELD NAME="timemodified" TYPE="int" LENGTH="10" NOTNULL="true" DEFAULT="0" SEQUENCE="false" COMMENT="time
this session was last used"/>
</FIELDS>
<KEYS>
  <KEY NAME="primary" TYPE="primary" FIELDS="id"/>
  <KEY NAME="scorm_data_id" TYPE="foreign" FIELDS="scorm" REFTABLE="scorm" REFFIELDS="id" COMMENT="scorm id"/>
</KEYS>
</TABLE>
```

Figura 45: Instal·lació taula adicional al mòdul SCORM

del formulari tal i com es veu a la figura 46.

```
class values_scorm_mod_form extends moodleform {
    public function definition () {
        global $CFG, $DB;
        $id = optional_param('id', '', PARAM_INT);
        $cm = get_coursemodule_from_id('scorm', $id, 0, true);

        if ($cm) {
            $sql = "SELECT * FROM {scorm_parameters_data} WHERE scorm = $cm->instance;";
            $result = $DB->get_record_sql($sql);
        }

        $mform = $this->_form;

        $mform->addElement('header', 'scorm_values_sco', 'Valors Scorm');
        $array_numeros = array(1,2,3,4,5,6,7,8,9,10);
        $mform->addElement('select', 'numero', 'Número', $array_numeros);
        if($result) $mform->setDefault('numero', ($result->number-1));
    }
}
```

Figura 46: Declaració classe del formulari, exemple recuperació BD i de input select

En el fitxer view.php, s'han de modificar diverses parts de codi. Per realitzar tota la funcionalitat actual on guarda els paràmetres scorm s'han realitzat canvis entre les línies 170 de codi i 362. Alhora d'afegir un nou paràmetre per monitoritzar s'hauria de veure totes les crides a la base de dades i veure com es guarda un atribut en un camp de la taula.

En la següent part de codi que es mostra en la figura 47), es comproven si existeix unes variables en uns camps de la base de dades per veure si podem afegir, i com, els paràmetres del formulari de la vista professor. La instanciació de un stdClass per després guardar la nova variable, que en aquest cas seria a sota del else, i si només l'actualitzéssim seria en aquesta part de codi.

Cal fer una menció especial a com es guarda el valor del paràmetre `çmi.suspend_data`, ja que les opcions que passen al Javascript, es guarden i es transmeten a través d'aquest paràmetre. Primer de tot, quan es va al formulari del professor les dades que es guarden allà van a parar a `mdl_scorm_parameters_data`, es guarda una configuració per mòdul activitat scorm instanciat a Moodle. Alhora de començar el joc, es recupera aquesta dada per passar-la al usuari. La implementació de Moodle llegeix directament els mètodes necessaris i la taula `mdl_scorm_scoes_track` per enviar els paràmetres guardats al Javascript, i per tant tots els paràmetres que es vulguin enviar s'han de guardar a `mdl_scorm_scoes_track`.

Com s'observa en la figura 48, alhora de guardar el `çmi.suspend_data`, el value d'aquest s'ha de guardar en tokens i separats per una coma. Si es desitja es pot separar els tokens d'una altra forma, però llavors es tindria que modificar les dades guardades i tot relacionat en aquesta taula i el paràmetre `çmi.suspend_data`. Es recorda que `çmi.suspend_data` és un paràmetre que tant llegeix com escriu i que per la nostra implementació només s'ha usat la opció de llegir, també es recorda que es un paràmetre que té molta capacitat d'informació arribant de 4kb a 64kb en les últimes implementacions de SCORM. Si es volgués usar el paràmetre `çmi.launch_data` també seria vàlid per aquesta implementació en comptes de `çmi.suspend_data`.

```

$result2 = $DB->get_record_sql($sql);

$sql2 = "SELECT * FROM {scorm_scoes_track} WHERE element = 'x.start.time' AND attempt = $attempt AND userid =
        $USER->id AND scormid = $scorm->id AND scoid = $scoid;";

$result3 = $DB->get_record_sql($sql2);

if ($result2 && !$result3) {
    $save = new stdClass();
    $save->id = $result2->id;
    $save->userid = $USER->id;
    $save->scormid = $scorm->id;
    $save->scoid = $scoid;

    if ($result3) {
        $save->attempt = $result3->attempt;
    } else {
        $save->attempt = $result2->attempt;
    }

    $save->element = 'cmi.suspend_data';
    $save->value = $result->number.", ".$result->signe.", ".$result->level;
    $save->timemodified = time();

    $DB->update_record('scorm_scoes_track', $save);
} else {

```

Figura 47: Exemple canvis nou atribut

```

$result2 = $DB->get_record_sql($sql);

$sql2 = "SELECT * FROM {scorm_scoes_track} WHERE element = 'x.start.time' AND attempt = $attempt AND userid =
        $USER->id AND scormid = $scorm->id AND scoid = $scoid;";

$result3 = $DB->get_record_sql($sql2);

if ($result2 && !$result3) {
    $save = new stdClass();
    $save->id = $result2->id;
    $save->userid = $USER->id;
    $save->scormid = $scorm->id;
    $save->scoid = $scoid;

    if ($result3) {
        $save->attempt = $result3->attempt;
    } else {
        $save->attempt = $result2->attempt;
    }

    $save->element = 'cmi.suspend_data';
    $save->value = $result->number.", ".$result->signe.", ".$result->level;
    $save->timemodified = time();

    $DB->update_record('scorm_scoes_track', $save);
} else {

```

Figura 48: Guardar paràmetre suspend_data a la taula mdl_scorm_scoes_track

A.2.2 Javascript

En el cas del Javascript, primer s'hauria de pensar quin paràmetre monitoritzar. Es podria monitoritzar per exemple les vides del breakout, donant la opció amb un input select des de Moodle. Per realitzar aquests canvis es tindria que canviar diverses opcions de Moodle esmentades en els paràgrafs anteriors i una petita part de codi del Javascript.

En el codi de Javascript ja es crida l'API Adapter i es criden diversos mètodes per iniciar, finalitzar, recuperar i enviar scores. Com s'ha mencionat en l'implementació de Moodle, el paràmetre `cmi.suspend_data` guarda la informació en la taula `mdl_scorm_scoes_track` en forma de tokens i separat per comes. Per tant, cal que la nostra implementació en Javascript reculli aquests paràmetres i faci el split en diferents variables. (veure Figura 49)

```
function calcScormValues () {  
  
    var suspend_data = retrieveDataValue("cmi.suspend_data");  
    // alert(suspend_data);  
    if (suspend_data.length > 0) {  
        var splitted_vars = suspend_data.split(",");  
        number = splitted_vars[0];  
        signe = splitted_vars[1];  
        level = splitted_vars[2];  
    }  
}
```

Figura 49: Recuperar el paràmetre `cmi.suspend_data` de Moodle amb Javascript

A.2.3 Unity

Igual que en el Javascript, per Unity cal pensar quina opció parametritzar per guardar a la base de dades de Moodle. Tal i com s'ha mencionat anteriorment, caldria realitzar uns canvis en el codi i base de dades de Moodle, i a més a més d'uns petits canvis al C Sharp de Unity.

En Unity, l'API es cridada en el mètode start del joc, al GameController, quan cridem al Initialize fa les búsquedes pertinents per trobar-la. Una vegada iniciat la comunicació amb el LMS, es fan les crides pertinents de SCORM per recuperar les dades de Moodle, i es fa la crida també de la funció `GetSuspendData()`, funció que es declara dins de GameController pero modularitzar aquesta funcionalitat, i es crida la funció `GetSuspendData()` de l'API. A partir d'aquí els passos són els mateixos que al Javascript, i es fa split amb les dades que es reben de `GetSuspendData()`, i es parametritzen en diferents variables (numero, signe i nivell) com es mostra a la figura 50.

```

void GetSuspendData()
{
    string suspendData = scormService.GetSuspendData();

    if (suspendData.Length > 0)
    {
        string[] split_suspendData = suspendData.Split(char.Parse(","));
        numero = int.Parse(split_suspendData[0]);
        signe = int.Parse(split_suspendData[1]);
        level = int.Parse(split_suspendData[2]);
    }
}

```

Figura 50: Recuperar el paràmetre “cmi.suspend_data” de Moodle amb Unity

B Manual d’usuari

Per configurar i afegir un nou joc a Moodle, cal tenir un usuari amb rol administrador o editingteacher. Una vegada s’ha iniciat sessió, es va al curs on es vol afegir un nou joc, i es farà click a la roda dentada on desplegarà un menú con el de la figura 51, per trobar i fer click a la opció (*Activar edició*) . Al activar aquest botó es podrà afegir una nova activitat si es fa click i on sortirà un pop-up amb un menú d’activitats. Cal seleccionar un tipus d’activitat SCORM com es pot observar a la figura 52, d’entre les que tenim al menú, i finalment s’ha de prémer el botó **afegir**.



Figura 51: Activar edició

En pulsar el botó **afegir**, s’entrarà dins d’un menú de configuració d’activitat SCORM. Si es fa una visió general de la configuració, es pot veure que demana bastant informació, però si el que es vol es realitzar una configuració ràpida, només es necessiten tres dades, nom de l’activitat, breu descripció d’aquesta i el més important, pujar un joc en format zip com es pot veure a la figura 53, on tingui un imsmanifest.xml l’arrel del directori, o directament pujar un imsmanifest.xml que per la nostra opció no té sentit. Per tant, s’escollirà la opció de pujar un joc

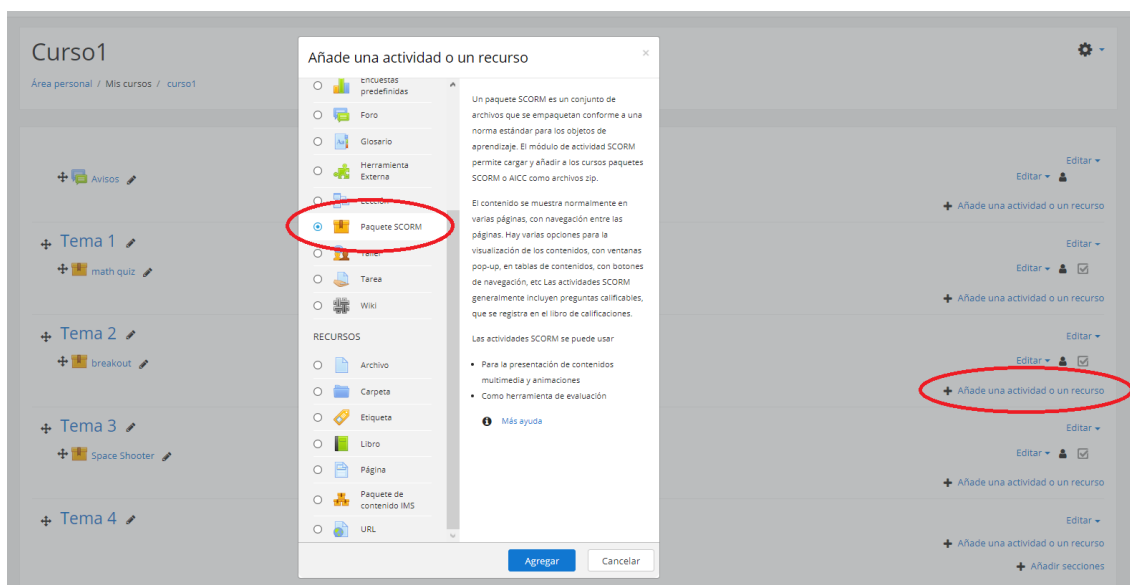


Figura 52: Seleccionar tipus d'activitat SCORM per afegir un nou joc

comprimit en .zip on l'arrel d'aquest arxiu existeixi un imsmanifest.xml i es guardin els canvis en qualsevol cas.

Una vegada guardades les dades, la configuració bàsica de l'activitat del joc s'ha acabat i ja es pot jugar al joc des de la vista de l'alumne. Encara que es pot configurar més paràmetres a l'activitat SCORM, alguns d'aquests poden ser interessants, a la figura 54 es mostren diversos paràmetres que es poden configurar.

Aparença Aquesta opció tracta amb temes de la visibilitat del joc SCORM, alguna de les seves opcions són tals com: canviar la opció de visibilitat en una finestra nova o mantenir-la en la actual, ocultar l'estructura del curs del reproductor o mostrar estat dels intents.

Disponibilitat Aquesta opció es interessant sobretot per deixar als alumnes un rang de dies i hores per realitzar l'activitat. La interfície és un petit formulari on es pot seleccionar el rang horari entre: disponible des de i disponible fins.

Qualificació Es pot modificar quina puntuació es vol guardar per puntuar a l'alumne, i seleccionar quina qualificació màxima es pot arribar.

Gestió d'intents Aquí es pot modificar paràmetres com el número d'intents, qualificació d'intents (més alta, promig, primer intent..), forçar nou intent o bloquejar després de l'últim intent.

Existeixen més configuracions generals d'aquesta activitat, però aquestes són les més interessants en relatiu amb el joc.

General

Nombre

Descripción

1

i

B

I

☐ Muestra la descripción en la página del curso

Paquete

Paquete

Archivos

Tipos de archivo aceptados:

application/xml .xml

Archivo (ZIP) .zip

Nunca

Puede arrastrar y soltar archivos aquí para a?

Actualizar frecuencia automáticamente

Apariencia

Disponibilidad

Calificación

Gestión de intentos

Configuración de compatibilidad

Ajustes comunes del módulo

Restricciones de acceso

Finalización de actividad

Marcas

Competencias

Guardar cambios y regresar al curso

Guardar cambios y mostrar

Cancelar

En este formulario hay campos obligatorios

Figura 53: Configuració activitat SCORM

Apariencia

Mostrar paquete

Ventana actual

Anchura

↓
100%

Altura

↓
500

Opciones (no admitidas por algunos navegadores)

☐ Permitir desplazamiento de la ventana

☐ Mostrar enlaces de directorio

☐ Mostrar la barra de ubicación

☐ Mostrar la barra de menú

☐ Mostrar la barra de herramientas

☐ Estatus

☒ Mostrar nombre de la actividad ?

Pasar por alto al estudiante la página de estructura de contenidos

! ?
Nunca

Ocultar botón de previsualización

! ?
No

Mostrar estructura del curso en la página de entrada

?
No

Mostrar la estructura del curso en el reproductor

! ?
Lateral

Mostrar navegación

! ?
Bajo el contenido

Desde la izquierda

!
-100

Desde arriba

↓
-100

Mostrar estado de intentos

?
Área personal y página de inicio

Ver menos...

Disponibilidad

Disponible desde

25
junio
2019
17
40

☐ Habilitar

Disponible hasta

25
junio
2019
17
40

☐ Habilitar

Calificación

Método de calificación

?
Calificación más alta

Calificación máxima

100

Gestión de intentos

Número de intentos

?
Intentos ilimitados

Calificación de intentos

?
Intento más alto

Forzar nuevo intento

?
No

Bloquear después último intento

?
No

Figura 54: Més opcions en la configuració d'activitat SCORM

Referències

- [1] http://publications.cetis.org.uk/wp-content/uploads/2011/10/WhatIsScorm2_web.pdf
- [2] <https://es.wikipedia.org/wiki/SCORM>
- [3] <https://www.easy-lms.com/es/centro-de-conocimiento/centro-de-conocimiento-lms/que-es-un-lms/item10182>
- [4] https://en.wikipedia.org/wiki/Learning_management_system
- [5] https://es.wikipedia.org/wiki/Tin_Can_API
- [6] <http://www.jsmadeeasy.com/javascripts/Games/Math>
- [7] https://developer.mozilla.org/es/docs/Games/Tutorials/2D_breakout_game_Phaser
- [8] <https://es.wikipedia.org/wiki/Arkanoid>
- [9] <https://unity3d.com/es/learn/tutorials/s/space-shooter-tutorial>
- [10] Advanced Distributed Learning Sharable Content Object Reference Model Version 1.2 2001 *The SCORM Run-Time Environment*
- [11] Cooking up a SCORM 2002-2003 *A SCORM 1.2 Content Cookbook for Developers*